This thesis presents a general approach to the validation of interprocedural data flow results for separated software modules, in order to enable the safe use of data flow results on devices which cannot afford to run the data flow analysis on their own. The underlying idea stems from the "Proof-Carrying-Code Principle", which utilises that it is easier to check the correctness of a given solution of a problem than to solve the problem.

The requirement to validate analysis results originally arose for Java Bytecode Verification on Smart Cards. The generalisation of this specific application to the validation of interprocedural data flow results enables advanced optimisations or security checks on limited devices in a scenario where the mobile code is transmitted via an inherently insecure transport media like the Internet. The validation ensures the correctness of the results but the code producer can perform the complex analysis on a more powerful machine.

The central contribution of this thesis is the extension of the validation approach to the interprocedural analyses and to separated software modules. This is vital in a mobile code scenario where different software modules can be dynamically loaded to the target device and where the potential interactions between the software modules and the runtime environment have to be considered.