

**Laud, Peeter**

**Secure implementation of asynchronous method calls and futures.** (English) [Zbl 1354.68034](#)  
Mitchell, Chris J. (ed.) et al., Trusted systems. 4th international conference, INTRUST 2012, London, UK, December 17–18, 2012. Proceedings. Berlin: Springer (ISBN 978-3-642-35370-3/pbk). Lecture Notes in Computer Science 7711, 25–47 (2012).

Summary: Programming languages suitable for distributed computation contain constructs that should map well to the structure of the underlying system executing the programs, while being easily usable by the programmers and amenable to computer-aided verification. For object-oriented languages, asynchronous method calls returning futures that will be filled only after the called method has finished its execution have been proposed as a reasonably simple and analyzable programming construct. In this paper, we show how to map from a language with asynchronous method calls and futures to a language with explicit communication primitives and cryptographic operations. Our target language is reasonably similar to common process calculi, and translating it further to e.g. the applied pi calculus requires only known techniques. The translation is valid even for programs executing in open environments, where method calls and futures can be transmitted between the program and the environment.

For the entire collection see [\[Zbl 1351.68025\]](#).

**MSC:**

**68N15** Theory of programming languages

**68N19** Other programming paradigms (object-oriented, sequential, concurrent, automatic, etc.)

**Software:**

[Creol](#)

**Full Text:** [DOI](#)

**References:**

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.