MRG: Mobile Resource Guarantees

www.lfcs.ed.ac.uk/mrg

The MRG project is building secure foundations for the next generation of mobile applications, using *proof-carrying code* to give mathematical guarantees of program safety.

The internet is now a major channel for software distribution: interactive web pages, automatic software updates; even complete applications and operating systems. All this is *mobile code*: fantastically convenient, but such a dynamic environment hugely magnifies the challenge of ensuring that software runs safely, securely and reliably.

When your broadband-connected PC brings together a mass of ad-hoc downloaded applications, from different places and authors, what can you know about how they will work together?

What do you let run on **your** phone?

The parallel growth of "smart" devices and appliances brings an extra dimension: with phones and portable music players, car navigation systems, games consoles and digital TVs all hooked up to the net, they too move to online software installation and upgrades. Mobile devices bring new challenges for mobile code providers:

- Tight limits on available resources — memory, processor time, battery life ...
- The executing environment itself is unpredictable — one user may download code to run on her phone; another the same code for his TV.

What tools can help application providers make sure their mobile code runs well anywhere? What can you do to be sure that downloaded software will run safely on your new mobile phone?

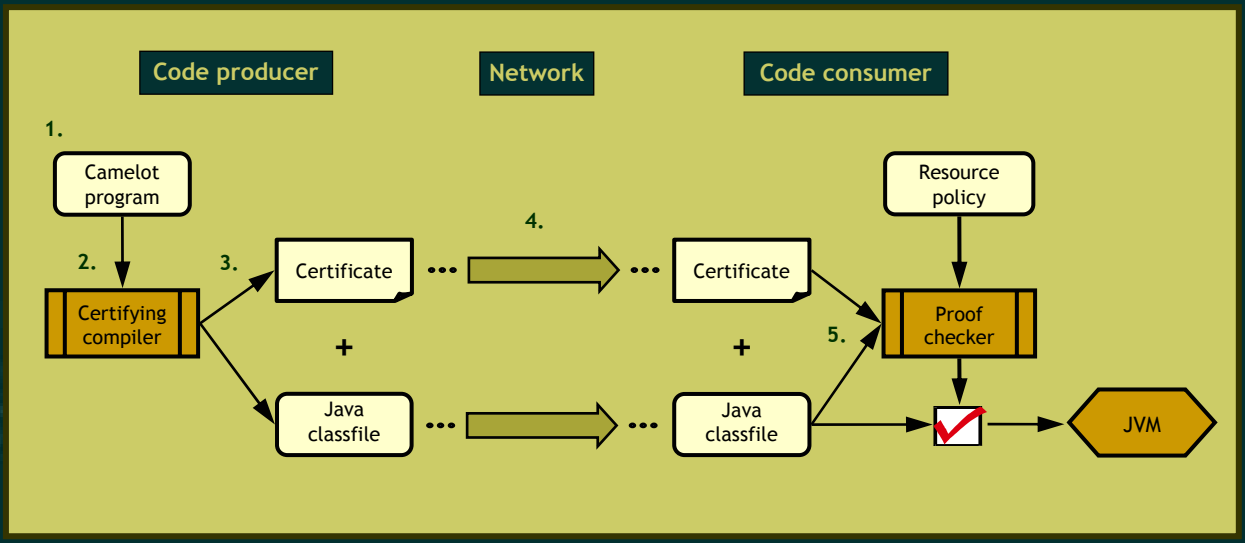
Strengthening trust with proven guarantees

The current trust standard for mobile code is a cryptographic certificate, a "signature" that identifies the code provider. This works well but has limitations: it requires a supporting infrastructure to distribute cryptographic keys; and although it tells us the origin of some mobile code, it can say nothing about the code itself.

A digital signature only tells you who to blame when things go wrong...

Mobile Resource Guarantees add an entirely new level of code assurance: programs carry with them a mathematical proof of their safety. These are inherently tamper-proof and unforgeable — they do not refer to an external authority but describe the code itself. Proofs are built by a *certifying compiler*, and any code receiver can automatically check the correctness of the proof before execution.

In MRG we apply this technology to guarantee bounds on the time and memory space used by programs running on the Java Virtual Machine. Code produced in the MRG system will execute as normal, but it also carries a proven resource guarantee. Code consumers can set a *resource policy* appropriate for their own Java device, and use an MRG agent to verify the compliance of downloaded code.



MRG technology

The diagram above shows how MRG combines a range of technological advances to provide guarantees for mobile code.

1. A programmer writes an application in the high-level *Camelot* language.
2. The Camelot *certifying compiler* analyses the program and generates executable Java bytecode together with a proof of its memory usage.
3. The proof certificate uses the MRG *bytecode logic*, which has been formally proved to give a complete and correct description of program behaviour.
4. Code and proof certificate travel together over the network.
5. An automatic proof checker verifies that the code meets the local *resource policy*, before safely executing it on a standard Java virtual machine.

There is an online demonstrator at the project site http://www.lfcs.ed.ac.uk/mrg, as well as downloads of all the MRG tools and documentation.

Future applications

MRG works today, to guarantee time and space performance of programs. For the future, three research projects are finding new applications for MRG technology:

- **ReQueST** — with massive eScience databases, it can be more effective to send code to data than try to bring the data to you; but how will database owners safely run this foreign code?
  http://www.lfcs.ed.ac.uk/request
- **Mobius** — Mobility, Ubiquity and Security: innovative trust management for global computing, where the resources can be network access, concurrency, and the secure flow of information itself.
  http://mobius.inria.fr
- **EmBounded** — with real-time requirements and tight platforms, embedded systems needs precise resource prediction and management; MRG code analysis can do this.
  http://www.embounded.org

All these projects combine industrial and scientific partners, exploiting the technology of mobile code guarantees in real-world applications.

Mobile Resource Guarantees (MRG) is a collaboration between the University of Edinburgh and Ludwig-Maximilians-Universität München, funded by the European Community's *Information Society Technologies* Programme (1998-2002) under the FET proactive initiative on Global Computing.