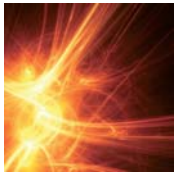


Reasons to Believe: Digital Evidence to Guarantee Trustworthy Mobile Code

Ian Stark

The University of Edinburgh

fet09 | 23 April 2009 | Prague

Mobius

Mobility, Ubiquity and Security

Enabling proof-carrying code for Java on mobile devices

http://mobius.inria.fr

Overview

Background

Digital Evidence and why we need it for global computers

FET Integrated Project

Mobius — Mobility, Ubiquity and Security

Challenges

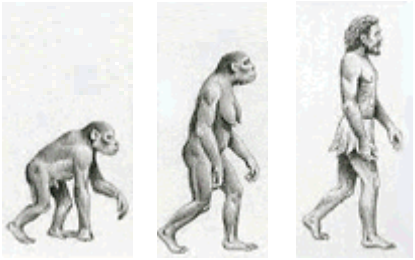
How can we create digital evidence? How can we use it?

Context: Global Computers

Mobile devices

Connected, adaptable, portable, personalized.

For now, smartphones. Eventually, everything.



Stages of sophistication

- Single-purpose, firmware controlled by manufacturer

  (TV, internet radio, camera, ebook, picture frame)

- Multi-purpose, software managed by manufacturer/distributer

  (MP3, portable media player, satnav, some mobile phones)

- General-purpose, software managed by user

  (PC, netbook, PDA, smartphone)

The View from Two Ends

User

- Download everything: documents, widgets, applications, libraries, OS extensions and updates.
- Everything must work together, and keep on doing so.

Software Developer

- Target environment heterogenous and highly unpredictable.
  - Varied devices, some in sandboxes, some virtual;
  - Different resource availability and demands;
  - Alternate OS/platforms and versions;
  - Must coexist with arbitrary other applications
- And do it all using minimal power and other resources.

In the middle…

The Rise of the App Store

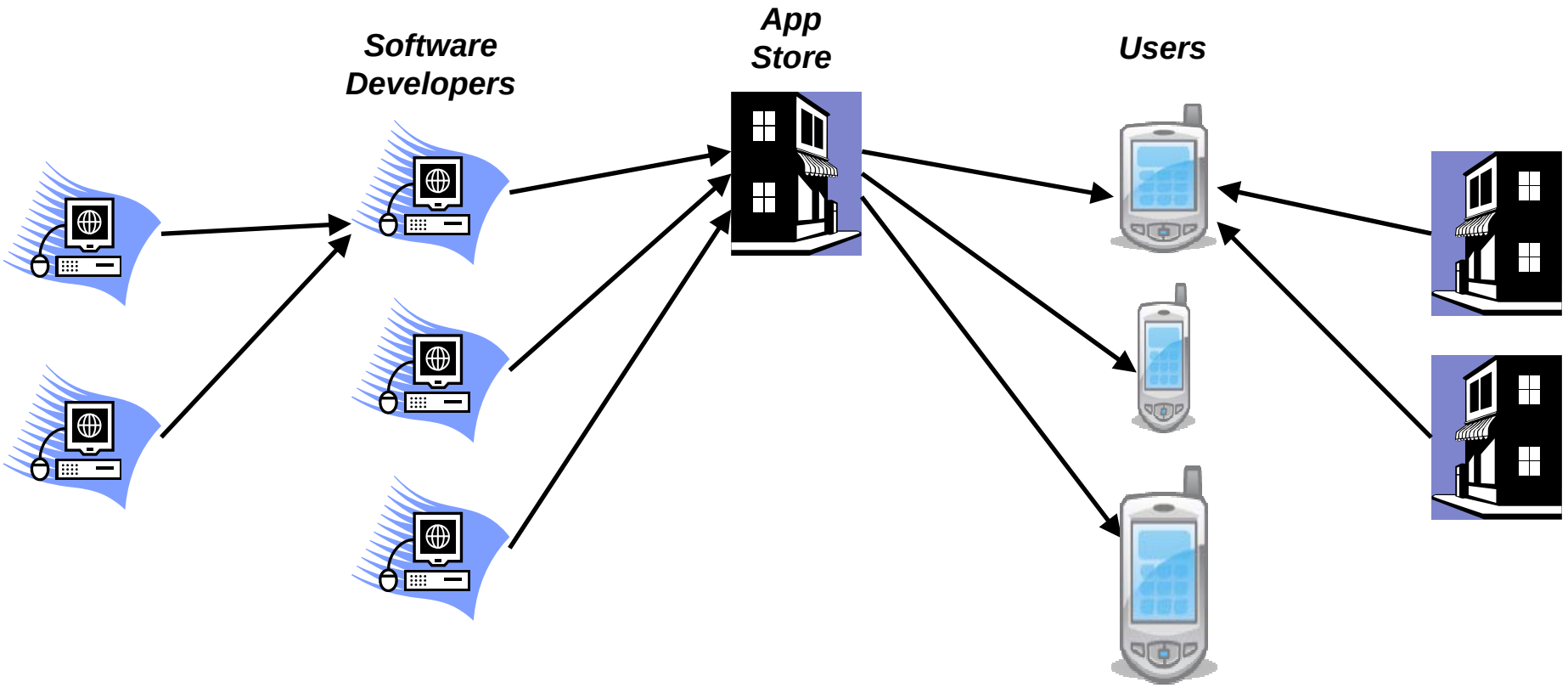
User gets Large numbers of applications, from multiple sources

Software developer gets Many customers, on many devices

The app store is a gatekeeper, and its success depends on trust:

- User trusts the store to provide worthwhile and safe applications at a reasonable price
- Software developer trusts in secure and reliable managed distribution with reasonable remuneration

Connections and Trust



This global network of mobile code depends on trust: built from reputation, and sealed in a hierarchy signed with cryptographic keys.

This is enough to trust the *identity* of others, and we might trust their *reputation*: but what does that tell us about the software itself?

Reputation is Only Part of the Answer

Reputation is not enough: we need information about the software itself, and how it will behave in different environments.

Varied devices No 3rd party iPhones (yet), but Windows Mobile and Android run on all kinds of devices, all with different capabilities. (memory, speed, power, graphics, network, ...)

Mixed ecosystem Software needs to coexist and interact with arbitrary applications, libraries, services, local and remote.

An app store also needs to convince software developers:

- What are its policies for acceptance?
- Are they transparent?
- Do they depend on the provider, or the software?

Rejected! 10 iPhone Apps That Didn't Make Apple's App Store
PC World 23 February 2009

Digital Evidence: Reason to Believe

We can strengthen trust in globally mobile code by supplementing digital signatures with *digital evidence:*

Digital Signatures

- Cryptographic, confirming that software has been approved
- Checked against external trust hierarchy of public keys

Digital Evidence

- Certificate presenting data about the software itself
- Confirms key aspects of software behaviour
- Can be independently checked, without external authority

Both rely on mathematical foundations to ensure that no certificate can be forged, and that code cannot be tampered with.

Both can work without revealing any original source code.

Varieties of Digital Evidence

Different kinds of evidence:

- Machine-checked proofs; fixpoints for abstract interpretation; loop invariants for bytecode logics; constraint solutions; …

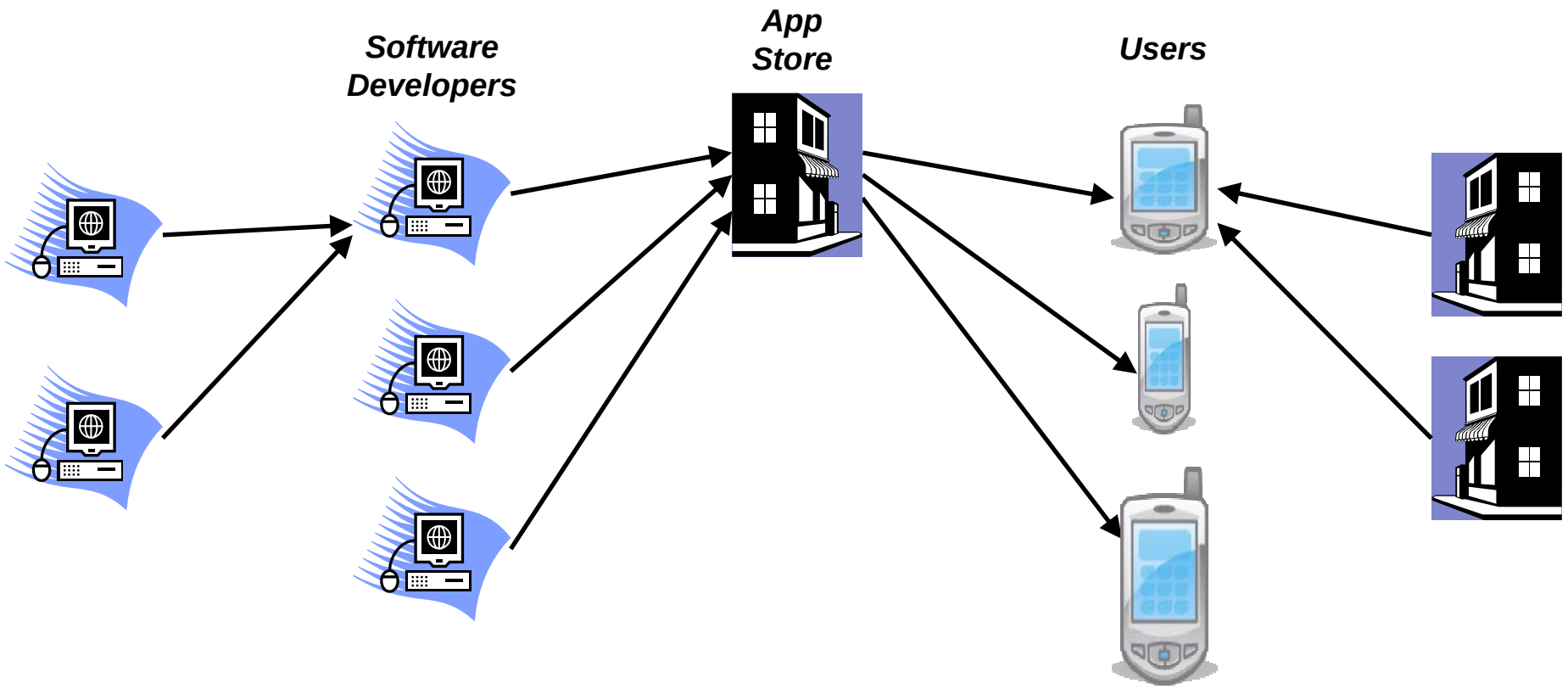
Different kinds of properties:

- Type safety, memory safety, stack and array bounds;
- Secure information flow, noninterference, declassification;
- Resource usage, access policies;
- Invocation protocols, method contracts; …

Existing examples:

- Stackmaps in the Java bytecode verifier; similarly for .NET;
- TAL in the *Singularity* operating system of Microsoft Research.

Using Evidence



Flows for Digital Evidence

- From store to user, evidence to satisfy security/resource policy
- From store to developer, stating objective acceptance policies
- From developer to store, providing evidence to meet these

Mobius

Mobility, Ubiquity and Security

Enabling proof-carrying code for Java on mobile devices

FET integrated project 2004–2009 with 16 partners in 10 countries, developing novel technologies for trustworthy global computing.

Started with a shared concept of *proof-carrying code,* extending this to gather in a range of code verification techniques: each creates digital evidence in certificates that guarantee program behaviour.

Concrete application domain: Java "midlets" on mobile phones.

Mobius Examples

Mobius base logic for Java Virtual Machine

Bicolano formalised JVM

Hybrid certificates: types+proofs

Generic resource logic

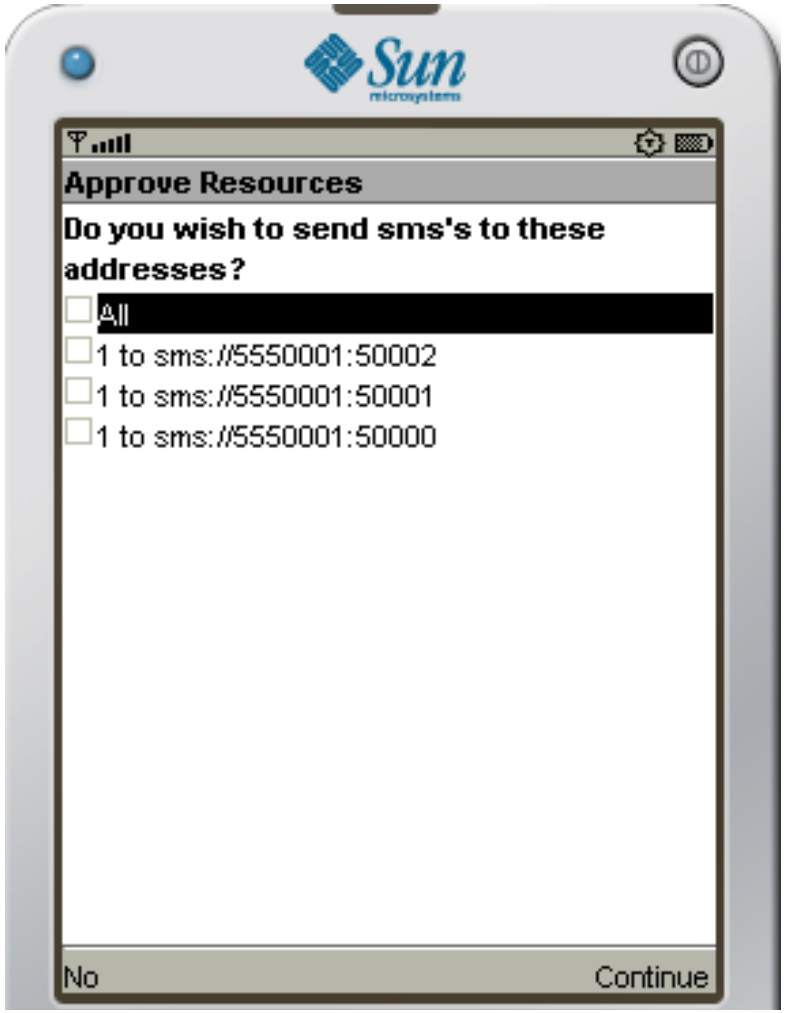
Amortised resource analysis

Platform-parameterised space and time analysis.

Certification of method protocols

Block-booking billable resources

Custom Eclipse environment with source and bytecode checkers



Mobius Examples

Certifying non-interference

Abstract interpretation to verify declassification of secure data

Relational reasoning for proofs of information flow

Formalisation of the relaxed memory model of multithreaded Java

Certificates for polyhedral inclusion

Lightweight certificates through computational reflection

Proof transformation for optimizing compilers

Summary

Context

- Mobile code for mobile devices: connected adaptable, portable, personalized
- All software from OS to app interacting and customized

Rise of the App Store

- Trust and reputation are central to global computing.
- Digital signatures tell us who code comes from; digital evidence can tell us about the code itself.

Mobius

- Developing technologies to create, transform, and check the digital evidence that provides proof of program behaviour

Questions and the Future

What else can be strengthened by digital evidence?

- Execution within databases; cloud computing; P2P computing?

Who can provide digital evidence, and how will they create it?

- Software authors, library writers, …
- Static analysers, optimizers, obfuscators?

Who can use digital evidence, and how will they check it?

- App stores: "wholesale"; users: "retail"; …
- On-device; offline; on-device; third-party validators.

Acknowledgements

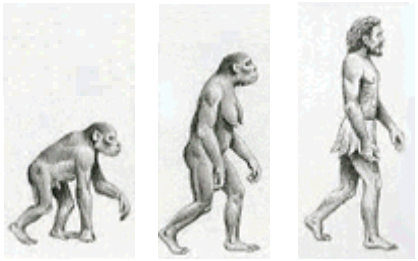
Mobius: Mobility, Ubiquity and Security

Mobius is funded from 2005–2009 as project IST-015905 under the Global Computing II proactive initiative of the Future and Emerging Technologies objective in the Information Society Technologies priority of the European Commission's 6th Framework programme.

Mobius Partners

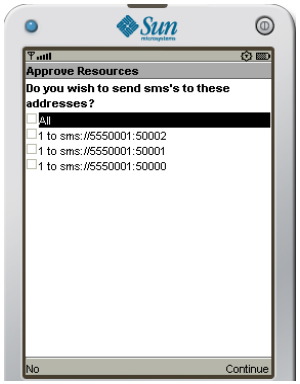
Universidad Politecnica de Madrid *(Coordinator);* INRIA, France; TLS Technologies, Poland; ETH Zürich; Radboud Universiteit Nijmegen; Ludwig-Maximilians-Universität, Munich; The University of Edinburgh; Institute of Cybernetics, Tallinn; Chalmers Technical University; Imperial College, London; University College Dublin; University of Warsaw; Trusted Labs, France; France Telecom; SAP AG, Germany; Technische Universität Darmstadt.

Picture Credits



March of Progress, p.2

Rudolph Zallinger. In *Early Man,* Life Nature Library, 1965.



Block booking applet, p.11

Andrew Todd and Patrick Maier
University of Edinburgh

On-device certificate checking, p.12

Pierre Crégut, France Télécom

http://www.youtube.com/watch?v=4AYiwo4NQtE