Bridging Global Computing with Grid (BIGG): Session on security and dependability

Gilles Barthe

INRIA, France

November 29, 2006

Setting

Huge distributed networks with conflicting needs/characteristics:

- *flexibility:* aimed at providing seamless access to located services,
- *heterogeneity:* devices may greatly vary in connectivity, computational power, libraries, etc.
- *extensibility:* possible to modify or enhance the computational infrastructure over the network (remote maintenance), or able to upgrade itself by fetching off-the-shelf components (self-healing or self-evolving system)
- *interactivity:* possible to delegate some tasks (computation, storing) to other devices
- *security:* devices and applications are subject to stringent constraints w.r.t. confidentiality, integrity, availability, privacy

Guaranteeing security and dependability: GC perspective

The result of a complex process that involves careful engineering:

- Develop computational models and programming languages that reflect/exploit the underlying infrastructure
- Define security goals, analyze threats and develop a security framework that enforces security goals (infrastructure security)
- Analysis and verification of security framework!
- Analysis and verification of programs (application security)

Security and dependability issues in GC

Traditional security architectures are not accurate:

- Computational model is evolving: distinction between applications and systems gradually disappears, thus more and more code will have an impact on security;
- Development model is evolving: code is increasingly developed through integration/evolution of components, hence implications of security mechanisms should be understood at a high level;
- Deployment model is evolving: code is deployed on heterogeneous devices that may have specific enforcement mechanisms (e.g. due to limited resources/connectivity);
- Security goals must be refined: security goals increasingly involve quantitative issues (amount of resource usage, information leaked or tainted, responsiveness). In addition, probabilistic guarantees are often more feasible than absolute certainty.

Strengths of the GC community

GC community is strongly influenced by pioneering work on:

- programming languages
- computational models
- formal methods

Rigorous methods are used to support the

- design
- modeling
- analysis/verification

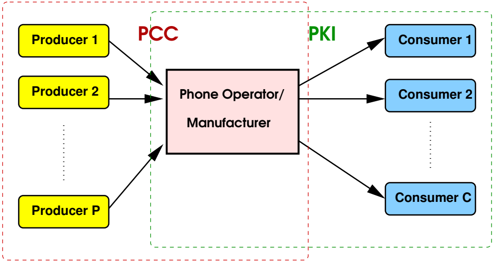
of secure software-intensive systems.

Security at work in GC

- Java-based mobile code:
  - strongly typed language with carefully crafted API,
  - access control via stack inspection (standard or history-based)
  - compile-time enforcement of resource and information flow policies for Java (developer perspective) and bytecode (consumer perspective)
- Process algebras:
  - static/dynamic enforcement mechanisms for rich policies in core calculi that support a rich theory of mobility
  - applications to cryptographic protocols and web services

Trust mechanisms in GC

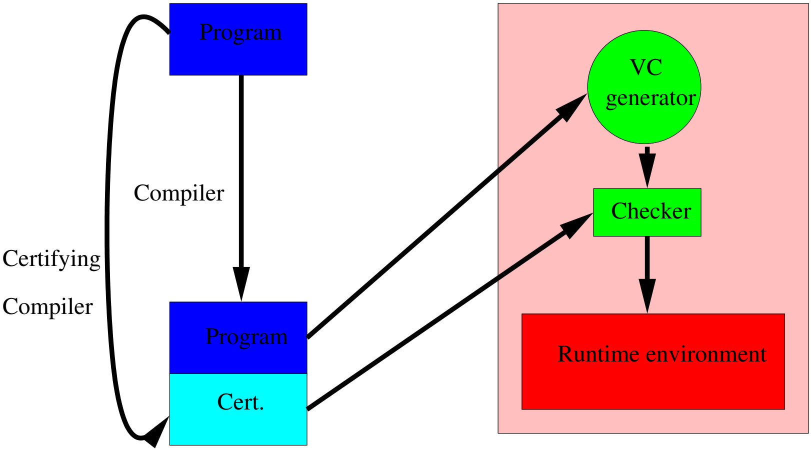
- trust by authority/reputation: based on standard PKI.
- trust by verifiable evidence: based on proof-carrying code
- combining trust is important: wholesale PCC



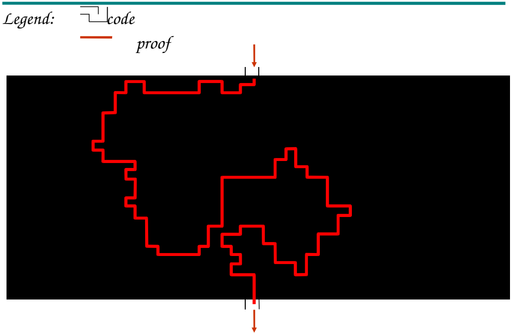
Challenges and opportunities for securing the grid: a GC perspective

- Formal models and policies (must account for distribution, fault-tolerance, use probabilities, etc)
- Language-based mechanisms
- Combination of trust mechanisms
- Securing remote software maintenance and evolution

Proof Carrying Code



- Condensed and formalized mathematical proofs which are self-evident, unforgeable, and straightforward to check.
- Proof checking $\neq$ Proof finding



Usage and challenges

Mobile code

- Usage: proof carrying code

  downloaded components come equipped with

  certificates
- Challenges:
  - Extend the scope of computational models and security policies enforced
  - Increase robustness and scalability through fundamental research in enabling technologies
  - Integration within mobile computing and component-based software systems.

Grid computing

- Usage: result certification

  result of computations come equipped with

  certificates
- Challenges:
  - Develop efficient checkers for computation-intensive problems through fundamental research in algorithms
  - Explore and quantify the role of partial checkers, e.g. probabilistic certificates.
  - Integration within Grid (and other) infrastructures

Acknowledgments

- Mobius integrated project within Global Computing II:

INRIA	*LMU Münich*
RU Nijmegen	*ETH Zürich*
U. Edinburgh	*Chalmers U.*
Tallinn U.	*Imperial College*
UC. Dublin	*U. Warsaw*
UP. Madrid	*RWTH Aachen*
SAP Research	*France Telecom*
Trusted Logic	*TLS*

- Beyond-The-Horizon project