Security and Trust issues for Services

Fabio Martinelli

Institute for Informatics and Telematics

National Research Council

(IIT-CNR)

Outline

- Brief overview of SENSORIA project (Software Engineering for Service Oriented Overlay computers)
  - Security and trust issues for services
- Specification&Modeling
  - Integrated security and trust modeling
- Analysis&Verification techniques
  - Model checking, type systems, control flow analysis, …
- Composition of services
  - Secure service composition by planning, …
- Conclusion

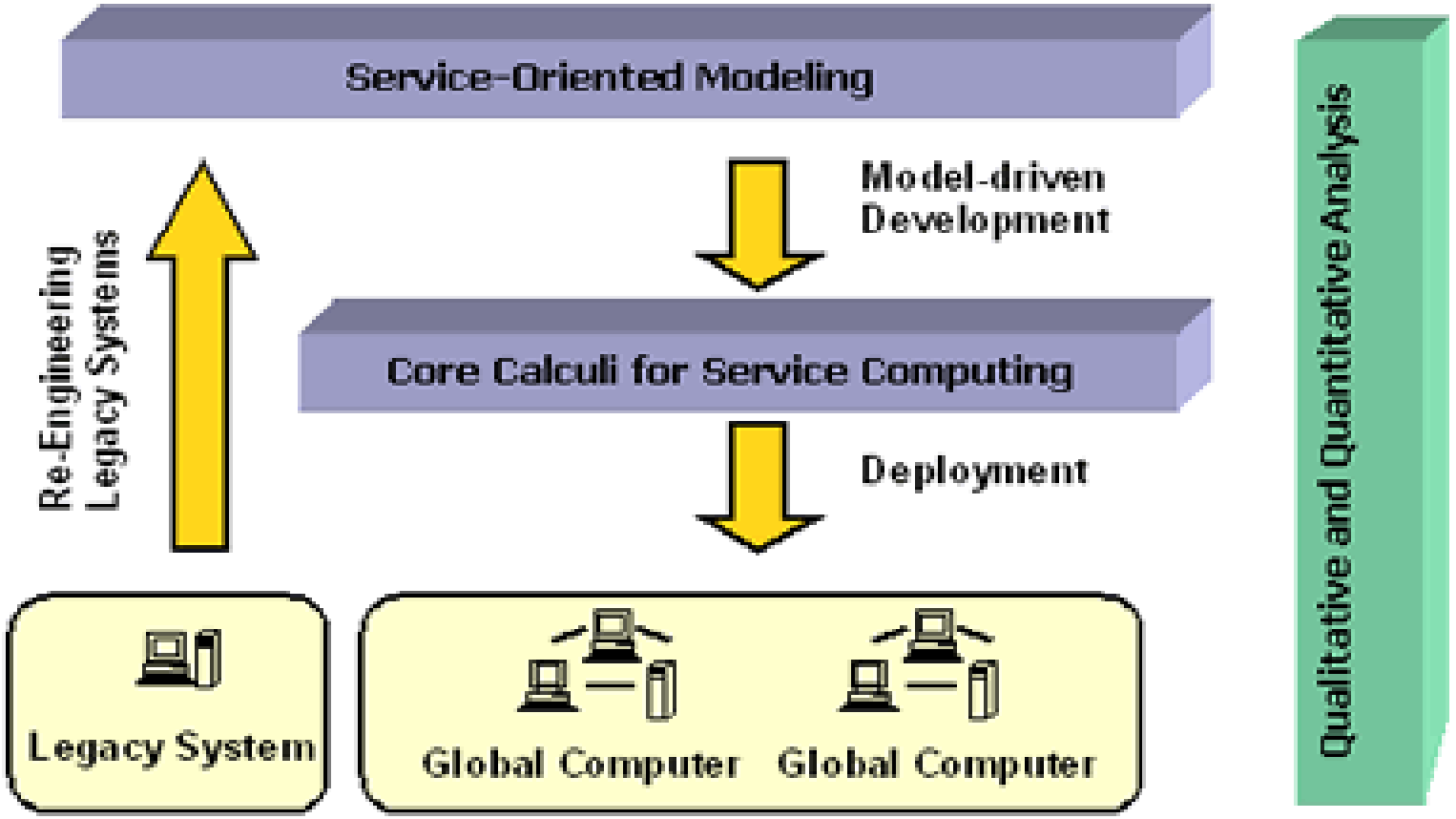
SENSORIA Goal

"… to develop a novel comprehensive approach to the engineering of software systems for *service*-oriented overlay computers where *foundational theories, techniques and methods* are fully *integrated* …"

Expected results:

- Language primitives for global service-oriented systems
  - modelling and programming
  - full mathematical semantics
- Qualitative and quantitative analysis methods for, e.g.,:
  - quality of service, security, performance and resource usage
- Sound engineering methods and deployment techniques
  - forward development through model-based transformations
  - re-engineering of legacy systems

In a picture


Service-Oriented Modeling
Model-driven Development
Re-Engineering Legacy Systems
Core Calculi for Service Computing
Deployment
Legacy System
Global Computer
Global Computer
Qualitative and Quantitative Analysis

Linguistic approach to services

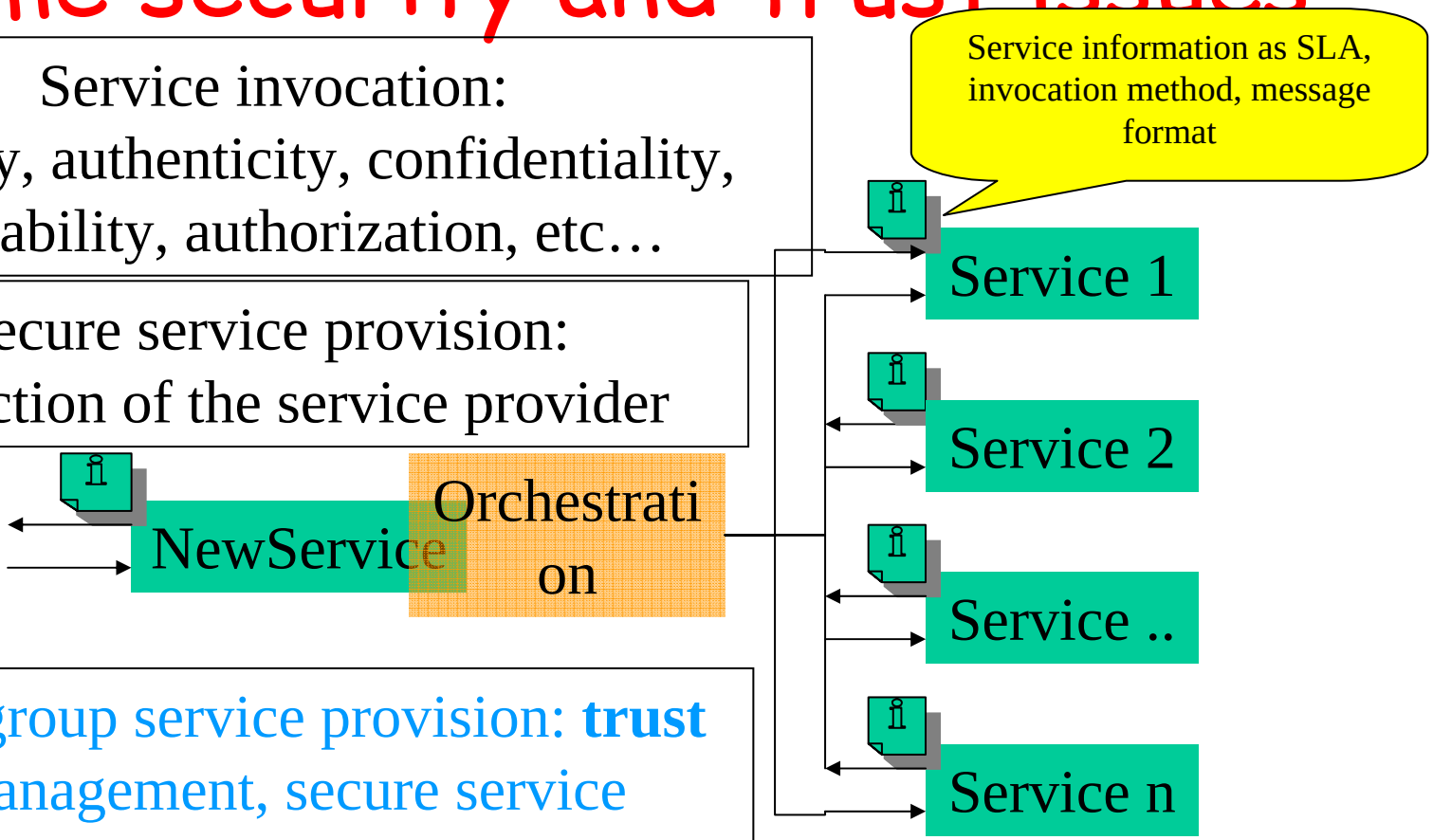
- **Service:**
  - autonomous, platform-independent computational entity that can be described, published, categorised, discovered
  - **Services** can be **dynamically assembled** for developing massively distributed, interoperable, evolvable systems and applications.
- Services:
  - are described by programs using specific constructs for service invocation, composition, execution compensation, etc …
  - Depending on the level of abstraction required:
    - UML to process calculi for distributed systems till implementation languages for service deployment

Some security and trust issues

Service invocation:
Integrity, authenticity, confidentiality, availability, authorization, etc…

Secure service provision:
Protection of the service provider

Secure group service provision: **trust** management, secure service composition, **privacy** policy management, SLA management, etc…



In particular …

- Services need to **interact**, i.e. using other services, and exploit on demand service composition:
  - Interaction may be based on trust relationships:
    - trust management mechanisms as trust formation, negotiation, monitoring, evoluation, etc…
    - reputation-based trust could also be applied
- Common **reputation/recommendation** mechanisms do not cope with malicius users in a satisfactory manner
  - Defame, collusion, free-riding, identity change, …
- Easing, monitoring and **securing** cooperation in service provision are mandatory:
  - Secure service coalition management
    - How would it be possible to identify not compliant services?
- **Privacy** issues about trust negotiation and reputation management
- …

Specification and modeling

- Development of formal languages for describing services with specific features for:
  - Cryptographic functions
  - Access control and privacy policies
  - Trust and reputation mechanisms
  - …
- This will allow to manage:
  - Application of Trust management concepts in Security, e.g.:
    - trust-based, on-demand, service provisioning
    - trust-based access control
    - …
  - Application of Security concepts in Trust and privacy management, e.g:
    - secure and reliable trust/reputation/recommendation mechanism
    - privacy issues in trust negotiation
    - …

Verification and analysis

- Several techniques are currently under investigation by the SENSORIA partners:
  - Model checking/module checking
    - based on exhaustive search of the computation space (either represented explicitly or by symbolic techniques). Module checking considers the situation where some components may be unspecified (e.g., unknown malicious components).
  - Type systems
    - based on (compositional) rules for assigning types to code. Types represent security properties of the code. It may impose programming disciplines.
  - Control Flow analysis (Nielson/DTU, Degano/Pisa)
    - algorithmic technique for checking properties of the code. Mainly it determines a safe approximation of the security property of interest, if the code satisfies this abstract property it also satisfies the concrete one (motto: err on the safe side)
  - …

Composing services by planning (Degano / Pisa)

- Services offer a "semantic" interface
  - abstraction of their behaviour
  - security guarantees

  and protect themselves from callers through their local security policies
- Clients invoke services "by property"
  - require a specific behaviour
  - impose their own policies the service has to respect
- Orchestrator
  - discovers services matching the requests
  - checks the security constraints are never violated
  - determine the correct execution plans

Technically:

- a Type&Effect system extracts the abstract behaviour
  - abstract behaviour are composed according to a plan and model-checked for checking that no security violation occurs at run time
- code may be annotated giving a "mixed approach:
  - static checks (T&E system + model-checking) plus run-time execution monitor

Conclusion

- Focus on integrated security and trust **design** and **analysis** for services described by means of suitable languages