The Challenges of Secure and Trustworthy Service Composition in the Future Internet

Kashif Kifayat – Liverpool John Moores University

Per Håkon Meland – SINTEF ICT
Juan Bareño Guerenabarrena – Atos Origin
David Llewellyn-Jones – Liverpool John Moores University

D.Llewellyn-Jones@ljmu.ac.uk
http://aniketos.eu

27 June 2011
Sixth International Conference on System of Systems Engineering (SoSE 2011)
Albuquerque, NM, USA
Contents

- Project overview
- Context
  - Future Internet
  - Services and service composition
- Composite trust and security
- The Aniketos platform
  - Design-time security specification
  - Contract verification
  - Run-time monitoring
- Conclusion

Box image by ba1969: [http://www.sxc.hu/photo/1301543](http://www.sxc.hu/photo/1301543)
Aniketos Project

- EU FP7 funded project
  - “The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant no 257930.”
- Started August 2010 running until February 2014
  - 42 month project
- Main objective
  - Provide service developers and providers with a secure service development framework that includes methods, tools and security services that supports the design-time creation and run-time composition of secure dynamic services, where both the services and the threats are evolving.
- Currently still at an early stage
  - See http://aniketos.eu for more info
Aniketos Consortium

- Athens Technology Center SA
- Atos Origin
- DAEM S.A.
- DeepBlue
- SELEX ELSAG (ex Elsag Datamat)
- Italtel
- Liverpool John Moores University
- National Research Council of Italy
- SAP
- SEARCH Lab Ltd
- Stiftelsen SINTEF
- Tecnalia Research & Innovation
- Thales
- University of Salzburg
- University of Trento
- Waterford Institute of Technology
- Wind Telecomunicazioni S.p.A.
Future Internet

- Focused on networked services
  - Services offered ‘in the cloud’
- Focus on compositional services
- Classic System-of-Systems scenario
- Move
  - From monolithic full-service stack suppliers
  - To dynamic services built using multiple services from multiple providers
- Technologies to support this already widespread
  - Web services, UDDI, OSGi, etc.
Compose Service Case Studies

Air traffic service pool

Future telecom services

eGovernance: Land buying

Photo by Joe Lipson, CC license
Recursive Services

- Need to support a ‘recursive services’ scenario
  - Using a service, don’t need to know (or care) whether it’s a single service or composite service
  - As long as the interface is the same

- Applies to developers and end users
  - Developer may compose a service from other services
  - Those other services may themselves be composite

- Need to know properties
  - Of composed service taken as a whole (i.e. viewed as single service)
  - Not of services it’s composed from
Trust and Security

- Represents a significant challenge for the Future Internet
  - Security problems will hinder acceptance
  - Future Internet introduces a number of new difficulties

- Security challenges
  - Services made up of other services
  - Service composition may not be obvious externally
  - Services provided by multiple providers
  - Widespread adoption means security must be clear for non-technical users
Composite Security

- Not just enforcing single security property on all services
  - Distributed services from multiple providers
  - Difficulty knowing if a policy is violated or not
  - Service developers use other services as part of a composition
  - Service providers agree to fulfil a customer’s policy
    - Need to know whether their service can fulfil it
    - Need to decide whether this is the case
    - Need to tools to determine security properties based on composition
Composite Trust

Services requires not just security, but also trust
- Service provider claims to fulfil a security policy
- How can a service consumer trust this?
- Need tools for trust and verification

Recursive services introduce
- Composite trust
- Chains of trust
- Requirements on careful attribution
  - Who’s trust rating should be affected if something goes wrong?
Aniketos Approach

- Make composite services able to establish and maintain security and trustworthiness
Aniketos Approach

- Make composite services able to establish and maintain security and trustworthiness
Aniketos Approach – Objectives

- Ensure and manage trustworthiness of interoperable and dynamically evolving services (through trust models and metrics)
- Develop integral framework providing methods and tool support for secure interoperable service development, composition, adaptation and management through concept of Security Engineering
- Define how to efficiently analyse, solve and share information on how new threats and vulnerabilities can be mitigated or how services can adapt to them
- Promote and contribute to best practices, standards and own certification work related to security and trust
- Demonstrate and evaluate practical use of security techniques, frameworks, patterns and tools in ordinary development of software and service with end-user trials
Aniketos Approach

**Developer**
- Design and offer composite services
- Ensure service security properties can be determined
- Match requested policy with offered contract

**Consumer**
- Ensure services are trusted
- Verify properties against requested security properties
- Identify monitoring requirements

**Runtime platform**
- Monitor and update trust
- Monitor security
- Identify new threats and vulnerabilities
Platform Overview

- This approach is reflected in the platform design
- Incorporates
  - Design-time support
  - Run-time support
  - Community support
- This presentation focuses on
  - Design-time security property definition
  - Design-time and run-time evaluation and validation of security properties and service behaviour
  - Run-time monitoring and evaluation
Key Concepts

- **Trust**
  - Used to determine whether offered security contracts are likely to be adhered to

- **Security**
  - Security requirements are defined by a security contract requested by the consumer, and fulfilled by a security policy agreed by the provider

- **Threats**
  - Threats define the context
  - Different security may be needed as new threats and vulnerabilities are identified
Design-Time Security Specification
Design-Time Security Specification

- **Purpose**
  - For end-users and service developers
  - To identify, understand and express required security properties offered by a composite service
  - To determine relevant threats

- **Incorporates**
  - Requirements specification
  - Threat analysis
  - Threat recommendation
  - Contract negotiation
Design-Time Security Specification

- Security requirements specification
  - Implemented using a socio-technical modelling tool
  - Extends the Secure Tropos requirements engineering methodology
  - This methodology models
    - Goals
    - Trust and supervision
    - Dependencies
    - Permission delegation
    - Tasks and resources
Design-Time Security Specification

- Requirements specification determines security policy
  - Used as basis for contract negotiation
  - Defines alerts needed based on threats

- Threat repository
  - Allows identification of relevant threats
  - Provides threat monitoring details
  - Provides countermeasure details
  - Practical approach to threat mitigation
    - Example attack tree

---

This model has been created by SINTEF as part of the EU project SHIELDS. The model is under a cc by-sa license. This attack tree is partly based on an SSH attack described by Gary McGraw: Risk Analysis: Attack Trees & Other Tricks, Dr. Dobbs Journal August 01, 2002.
Contract Verification

- Threat response recommendation
- Trustworthiness predication
- Security property determination
- Security contract
- Composition plan
- Service security properties
- Trust
- Countermeasure compliance
Contract Verification

- **Purpose**
  - When a user is presented with a service
  - To allow the user to determine
    - Claimed security properties of service
    - If service can be trusted to fulfil them

- **Incorporates**
  - Property determination
    - Simulated/closed environment testing
    - Formal service verification (for code)
    - Formal protocol composition analysis
    - Pattern-based secure component composition analysis
  - Trust management
Contract Verification

- Security by Trust
  - Analysis/testing is time-consuming
  - Where trust is sufficiently high
    - Accept claims made by service provider without verification
    - Apply run-time monitoring and enforcement

- Security verification
  - Formal
    - Use formal techniques for contract composition
  - Testing
    - Run services in simulated environment to determine behaviours
  - Heuristic
    - Use known security patterns to ensure requirements are met
Run-Time Monitoring

- Service threat monitoring
- Security policy monitoring
- Monitor trustworthiness
- Notification module
- Contract violation alert
- Trustworthiness prediction
Run-Time Monitoring

- **Purpose**
  - To monitor actual service properties against claimed properties
  - To prevent security violations
  - To update trust where violations occur
  - To react to changes in the environment affecting security

- **Incorporates**
  - Security policy monitoring
  - Trustworthiness monitoring and feedback
  - Service threat monitoring
  - Security re-validation in case of changes
Run-Time Monitoring

- Threat monitoring
  - Based on threats identified at design-time
  - Threats extracted from threat repository

- Security policy monitoring
  - Monitoring of dynamic behaviour
  - Policy Enforcement Points
  - Changes to service structure

- Trustworthiness monitoring
  - Trust changes originating elsewhere affect contract
  - Policy violations impact on trust
Conclusion

- Aniketos platform to be implemented as services

Current progress

- High-level design of service structure completed
- Detailed designs of design-time platform completed
- Implementation of design-time services started

Platform incorporates

- Specification and checking of composite security properties at design-time
- Tools for developing contracts for composite services
- Ability to ensuring contracts match actual properties at design-time and run-time
- Run-time monitoring capabilities for security and trust management

More info

- Website: http://aniketos.eu
- Email: D.Llewellyn-Jones@ljmu.ac.uk