The Aniketos platform

Per Håkon Meland[1], Erkuden Rios[2], Vasilis Tountopoulos[3], Achim Brucker[4]

[1]SINTEF ICT, N-7465, Norway
per.h.meland@sintef.no
[2]TECNALIA Research & Innovation, Parque Tecnológico de Bizkaia 700, Spain.
erkuden.rios@tecnalia.com
[3]Athens Technology Center S.A., Halandri, Athens, Greece
v.tountopoulos@atc.gr
[4]SAP AG, Vincenz-Priessnitz-Str. 1, 76131 Karlsruhe, Germany
achim.brucker@sap.com

Abstract: The overall objective of Aniketos has been to help establish and maintain trustworthiness and secure behaviour in a constantly changing service environment. The resulting Aniketos platform contains existing and newly developed technology, methods, tools and security services that support the design-time creation and run-time dynamic behaviour of composite services, addressing service developers, service providers and service end users. This chapter gives and overview of the Aniketos platform as a whole and its software packages.

Keywords: Aniketos, platform, service composition, components

1 Introduction

The Future Internet will provide an environment in which a diverse range of services are offered by a diverse range of suppliers, and users are likely to unknowingly invoke underlying services in a dynamic and ad hoc manner. Moving from today's static services, we will see service consumers that transparently mix and match service components depending on service availability, quality, price and security attributes. Thus, the applications end users see may be composed of multiple services from many different providers, and the end users may have little in the way of guaranteeing that a particular service or service supplier will actually offer the security claimed.

We can illustrate service composition through a practical service example: Let's say a service developer wants to create a travel agency service, MyPerfectTravel, which lets the end user read about various destinations, check the weather forecast and book a trip containing flights, hotel, a hire car and tickets to leisure activities. Under the hood of such a composite service there will be a range of services from independent providers, but the end user only has to relate to MyPerfectTravel. There are many services like this already today, but these are largely static affairs, and require redesign once

© Springer-Verlag Berlin Heidelberg 2011

there is some sort of change. New and emerging technologies for dynamic service composition allow a more autonomous and ad-hoc approach, so that a service can adapt to another configuration at runtime. Let's say that MyPerfectTravel ordinarily uses a service from the Amadeus reservation system to handle the flight bookings, but this service becomes exposed to a threat, lowering its assurance level or availability. MyPerfectTravel should consequently react to this, and for instance replace this service component with a similar one from e.g. Galileo, Worldspan or Sabre without the end user ever noticing it.

The main objective of Aniketos is to establish and maintain trustworthiness and secure behaviour in a constantly changing service environment. Aniketos provides methods for analysing, solving, and sharing information on how new threats and vulnerabilities can be mitigated. We have constructed a platform for creating and maintaining secure and trusted composite services that:

- Complements state-of-the-art service composition technology. The platform provides methods, tool support and community services to support service implementation, discovery, composition, adaptation and management through the concept of full life-cycle security engineering.
- Allows definition, validation and monitoring of trustworthiness and security properties of composed and dynamically evolving services through models for requirements specification and business-processes enhanced with security policies and metrics.
- Makes it possible to efficiently analyse, solve and share information on how new threats and vulnerabilities affect the composition and can be mitigated, so that composed services can automatically adapt to them without loosing availability and end user trust.
-
- Manages to handle trustworthiness and security of adapted/recomposed services from a socio-technical perspective. Security and trust are not only a technical issue for the heterogeneous nature of composite services, but rather an interleaving problem between technical and social aspects that cannot be considered in isolation.
- Has been proved useful in business case studies showing its practical usability, addressing real end user needs, acceptance and trust of new composite services.

In section 2 we give a high level overview of the Aniketos platform, the processes, activities and stakeholders it supports. Section 3 gives more detail on the component structure and explains how they are grouped in different packages. Concluding remarks are given in section 4.

2 The Aniketos platform at a glance

The Aniketos platform complements existing state-of-the-art Service Oriented Architecture (SOA) frameworks by connecting emerging technological solutions with the human practices that are needed to create and maintain secure and trusted composite services. Figure 1 shows how Aniketos supports activities within the composite service

development and operation life circle. Service developers compose services at design-time by discovering and including available service components from external service providers. A composite service is only as strong as its weakest link, so the service developer uses Aniketos to make sure that service providers can be trusted and that service components do not unintentionally violate security policies through their interaction. A service component needs to provide an abstraction of its behaviour and security guarantees. Consuming composite services requires a specific behaviour and impose their own policies that the service components have to respect.

At runtime, the service provider offers the composite service to a service end user. Due to the Aniketos capabilities, this end user is able to put his trust in one party instead of relating to all of the service providers involved. He would like to have stable level of trust, being indifferent to whether service components are changed or service providers come and go. Aniketos helps the service provider perform intelligent adaptation or recomposition of the service, triggered by changes in component behaviour, change in the trustworthiness of a service provider, new threat information, and also changes in the operating environment (e.g. a service is being used for a purpose it was not originally intended for or the end user relocates to a more hostile environment).

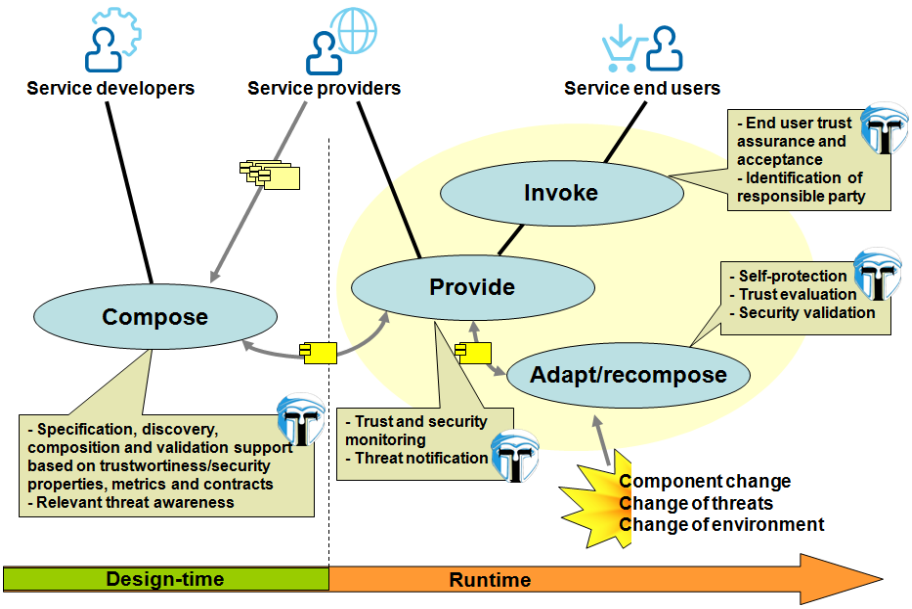

Figure 1: The composite service life cycle

As shown in Figure 2, the Aniketos platform itself can be structured into the three areas related to design-time support, runtime support and community support. These are explained in the text below in relation to different stakeholders.

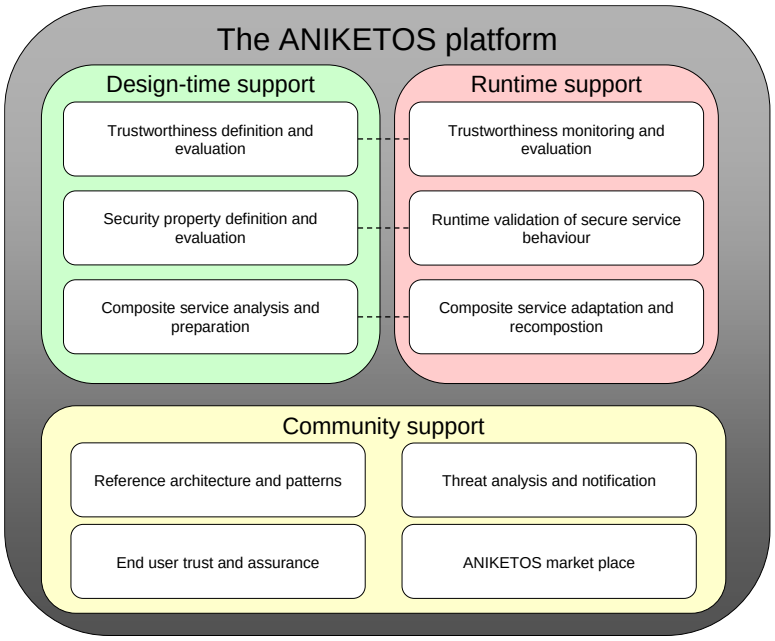

Figure 2: Overview of the Aniketos Platform.

2.1 Design-time support

This area consists of methodologies and tools that define and evaluate trustworthiness and security properties over and between external service components. This allows service developers to perform service discovery and composition based on security properties and metrics, not just functional descriptors. They are also able to choose service providers and service components by trustworthiness aspects for service composites. Composite services are analysed and prepared through automated on-line mechanisms that gather data concerning both individual components and service compositions as a whole, and the developer is informed about known threats to these through the threat notification from the community.

Aniketos did not set out to create a whole new process of developing composite services; a lot of work has already been done in this field and should therefore be exploited. Figure 3 shows typical work processes related to design-time service composition. There are other variations of this figure, with more/fewer process boxes (e.g. testing has been omitted since we focus on validation at design-time) and where the order might be a little bit different (e.g. contracts can be established after service assemble), but we think this one is a fairly generic version to which we can relate to. Note that loops have been omitted (e.g. if validation fails it will be necessary to go back one or several steps).

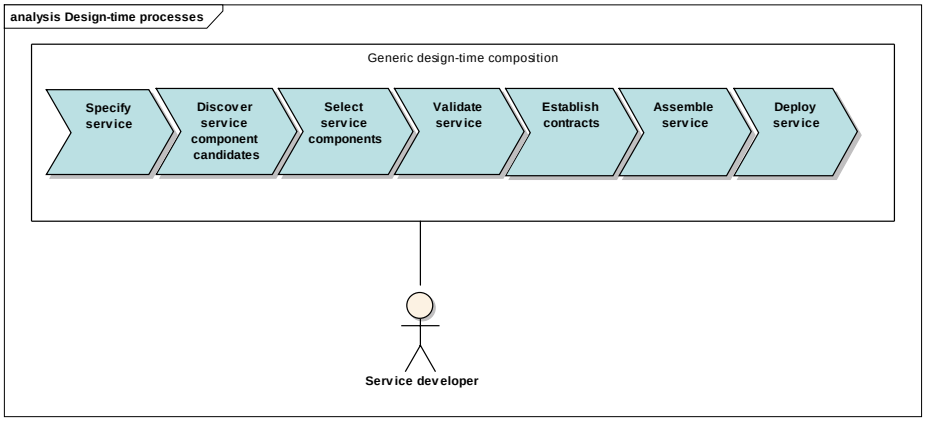

Figure 3: Typical processes related to design-time service composition

Only the service developer stakeholder has been included here, but this one can represent other more specific roles, such as service composer, service designer and service implementer. Additionally, the service owner and the service end user would typically be involved in giving high-level requirements input to the service specification process.

2.2 Runtime support

Any service provider must expect changes related to external dependencies, such as updates or alterations of service components, or unwanted circumstances that influence the service compositions. The design-time definitions are used to monitor and evaluate the trustworthiness and security violations of service components. The platform allows a proactive increase in trustworthiness by asking for more credentials and tries to control the damages in case of attack by selecting the appropriate security level on which the service can run. A runtime threat alert-and-adapt mechanism is able to receive emerging threat notifications from the community. All these are possible triggers to dynamic adaptation or recomposition of the service.

Figure 4 gives a generic overview of the runtime domain. The Provide service process is continuously running in the environment, and will at some point in time receive an alert from the Aniketos platform. This will indicate to the service provider that a service validation would be a wise thing to do. The service validation can have three outcomes:

1. The service is OK and the alert was nothing to care about, go back to regular provision.
2. The service is not OK, try and adapt with a reconfiguration (meaning keep the same service components but with some modifications).
3. The service is not OK, try to recompose (replace service components).

In the two latter cases the service provider would normally do a new validation since there has been a change.

In the lowermost part of the figure we have showed that monitoring is also something that is continuously done in the environment by the service provider. If something out of the ordinary is detected, an alert can be sent to the Aniketos platform, which would route this message to the relevant receivers. For instance, if a service provider detects an intrusion he would have to notify consuming composite services if this is a contract requirement.

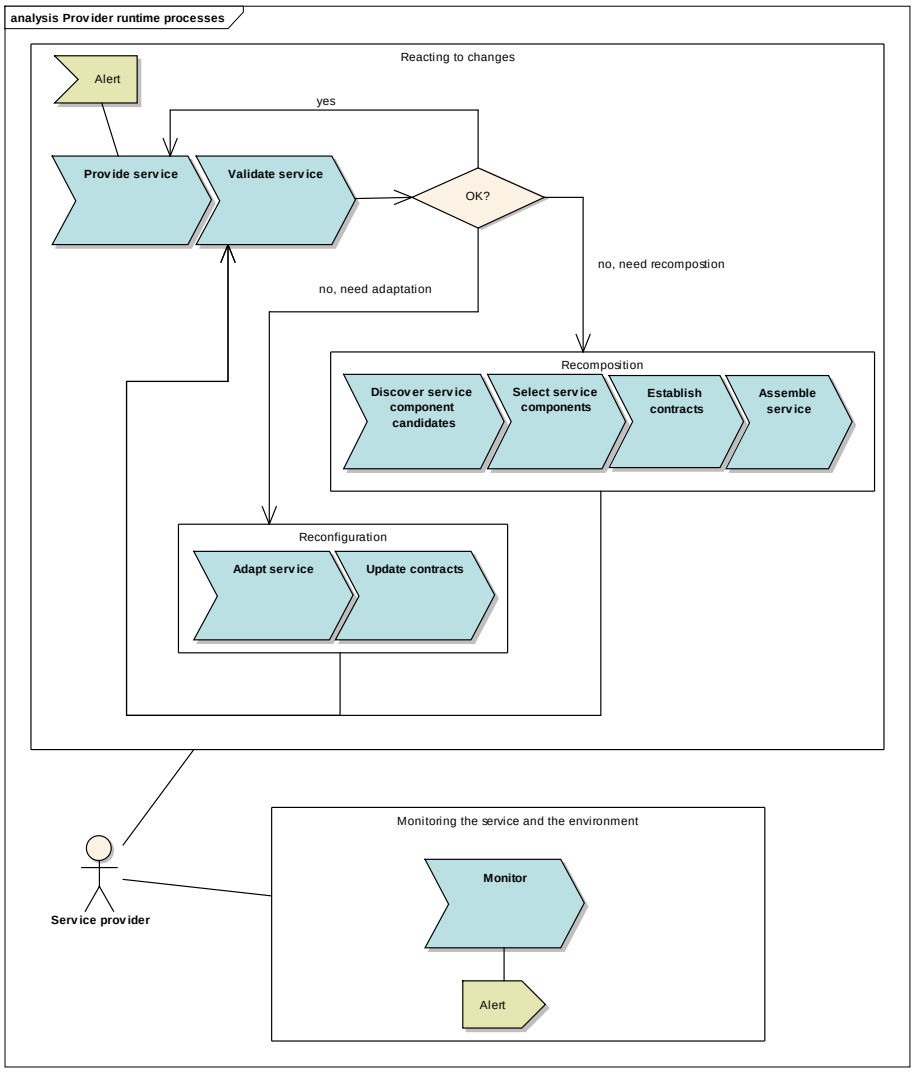

Figure 4: General processes related to runtime reaction to changes and monitoring

2.3 Community support

Service developers can find information on how to apply Aniketos as a part of the community support, which also includes example services, demonstration material, tutorials, process descriptions, development patterns and guidelines. Threat/countermeasure information and notifications are provided to both service developers and service providers in order to guide design-time composition or trigger runtime adaptation/recomposition based security goals or service components included in the composite service. The service end user will only need to relate to one entity that she can place her trust to and keep responsible in case something goes wrong, though a composite service has many underlying service providers. The Aniketos marketplace offers a way of requesting/offering service components with defined security and trustworthiness properties.

2.4 Overview of stakeholders

As already shown in the process diagrams, Aniketos supports several types of stakeholders who in one way or another are influenced by the platform. Figure 5 presents a more complete stakeholder breakdown (the arrow relationship between stakeholders is always inheritance/specialisation) along with a brief comment explaining the typical characteristics. These stakeholders are not mutually exclusive, for example a service provider might also be a service owner.

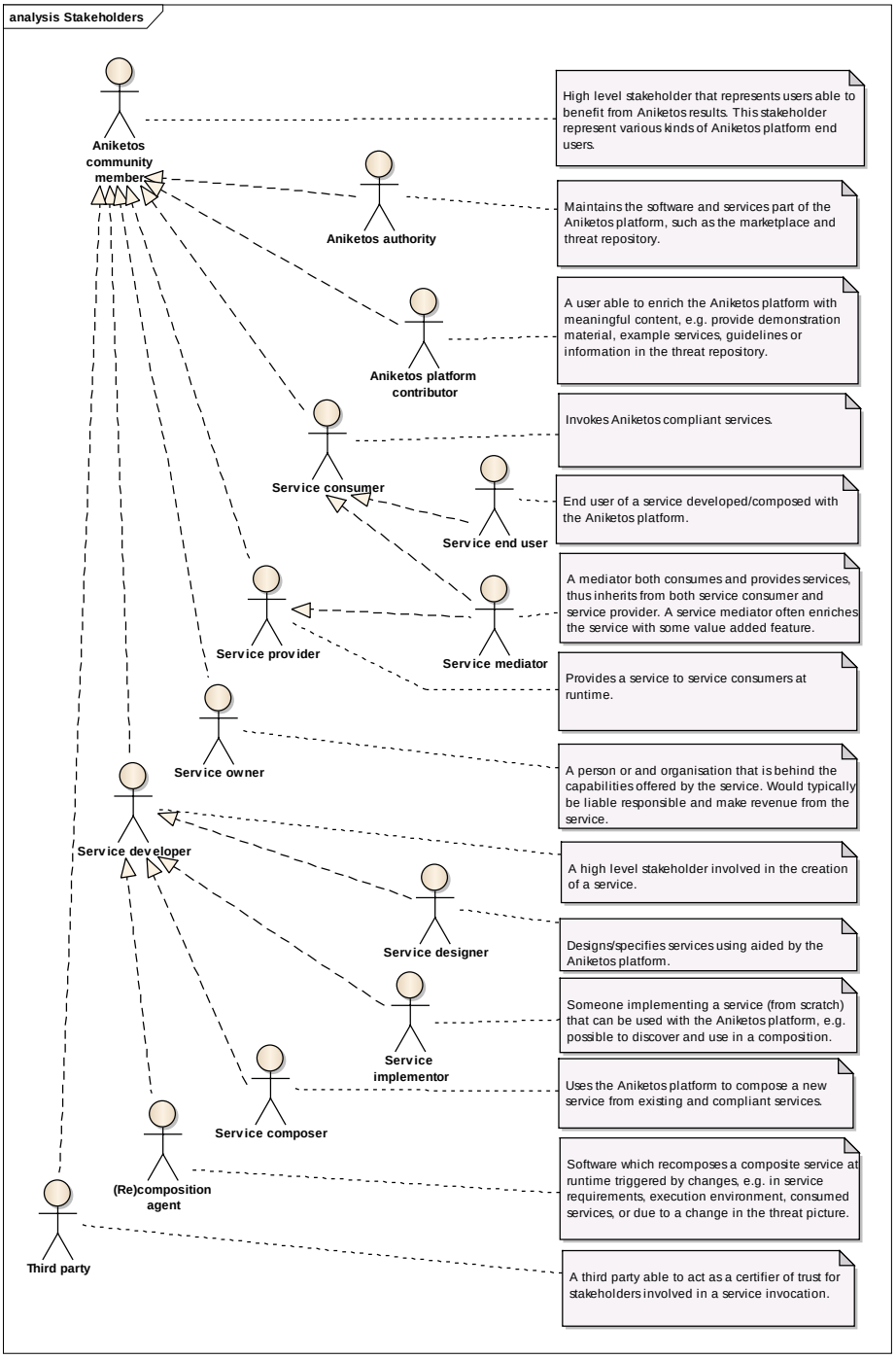

Figure 5: Stakeholders

3 Aniketos components and packaging

The Aniketos platform has been realised by a set of loosely coupled components that can be glued together based on what type of support is needed. Figure 6 gives a three-layered overview of these components, and we have used a colour convention to distinguish between the Aniketos platform (light yellow) and environment (light blue) components. An environment component is basically a reference implementation of something that interacts with Aniketos, and can be replaced by other tools that perform similar tasks. These tasks are the design and deployment of composite services (done through the Service Composition Framework - SCF), the execution and adaptation of service compositions (performed by Service Runtime Environment – SRE), the detection of deviations (through the Service Monitoring Module – SMM) and management of identity identification (through the Identity Management Service – IdM).

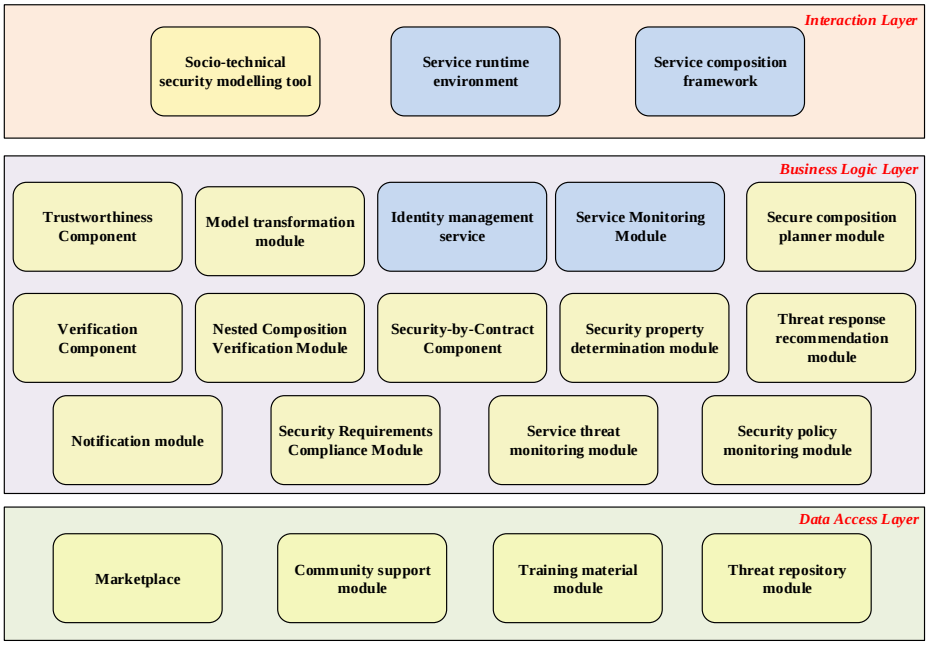

Figure 6: The layer-based conceptual representation of the Aniketos architectural design

The Aniketos Platform and Environment components have been grouped into four software packages, which better facilitate the delivery of the Aniketos platform functionalities to the target user groups. Below we describe the components from Figure 6 in relation to these software packages.

3.1 Socio-technical Security Requirements package

Description of the offered functionality
This package offers the functionality for a User friendly Interface to model the security relationships among resources mainly representing agents, roles, their security goals and the documents involved in the goal achievement (either as requirements or derivatives of the goal). The package offers four different views, namely:

- Social view: represents actor intentionality and sociality;
- Information view: represents the information in the considered organisation/setting together with the documents that represent such information, as well as the relationships among different pieces of information (documents).
- Authorisation view: represents the authorisations granted by some actors to other actors concerning the exchange and manipulation of information for particular purposes.
- Security Requirements: represents the list of security requirements expressed in terms of commitments that hold/should hold between actors to cover the security needs expressed in the above three views.

It also enables users associate the resources with potential threats, which express the vulnerabilities along the relationships among the defined resources. For each goal, a list of threats can be identified, which are followed from certain countermeasures consulting on the way the threats can be effectively addressed and risks can be mitigated.

The package can be used to model the security requirements of a holistic application and/or parts of it. Of particular interest is the fact that this package can model the security requirements for a service composite process, which has to be developed from scratch or already exists and needs to conform to specific security rules.

Involved Aniketos Components
This package consists of the following Aniketos components and modules:

- The Socio-Technical Security Modelling Tool (STS), which offers the main tool to model security requirements.
- The STS Threats Plugin, which enables attach threats and respective countermeasures to the security goals of a composite service.
- The Threat Repository Module (TRM), which exposes the list of registered threats and countermeasures encountered in ICT systems.

Main Outcome

- A document (in pdf) of the security relationships among the agents, the roles, the goals and the involved documents.
- An XML file with the set of security requirements specifications applied to the defined goals, expressed as commitments between the actors.

3.2 Secure Service Specification and Deployment package

Description of the offered functionality

This package enables modelling the process of a composite service and configuring the security requirements with respect to the components comprising the composite service specification. It also enables to deploy a composite service specification, which is extended with security properties by offering management functionalities for the maintenance of a list of services, which are described, based on their security properties. For each service, the functional specification is enriched with security characteristics, detailing the level of security that can be supported by the specific service.

The package offers the possibility to publish Aniketos compliant services to a service registry and supports searching in this registry to discover the most appropriate service descriptions, which comply with a set of security requirements.

Currently, a set of security properties are supported, which have to do with the trustworthiness, the separation and binding of duty, the confidentiality, the non-repudiation and the integrity.

Involved Aniketos Components

This package consists of the following Aniketos components:

- The Model Transformation Module (MTM), which can be optionally used when it is required the transformation between the security requirements, as expressed (with STS package) in the form of commitments, and the formal service specification. This module can also create the template for the consumer's security policy (in Conspec format), based on the commitments. Finally, the module can help in adding or completing security information of an existing service specification with the security requirements expressed as commitments.
- The Identity Management Service (IdM), which offers authentication and authorisation services and feeds the package tools with the roles, which should be defined in the service design.
- The Security Requirements Compliance Module (SRCM), which can be optionally used to compare the service specification created from MTM and the security requirements specification and verify that these two are compliant with each other.
- The Service Composition Framework (SCF), which is a process modelling tool (based on Activiti Designer) and is used to model the composite service processes. This component acts as an orchestrator between the various components of this package.
- The Threat Repository Module (TRM), which exposes the list of registered threats and countermeasures encountered in ICT systems.
- The Threat Response Recommendation Module (TRRM), which provides suggestions on the countermeasures, which should be applied during service composition in order to deal with those threats related to the development and deployment of a composite service.

- The Composition Security Validation Module (CSVM), which is part of the Verification Component and enables define security properties with respect to the separation or binding of duty and provides static analysis of the defined security requirements with respect to the specified composite service processes.
- The Conspec Editor, which is a flexible UI tool embedded in SCF to help security experts defining properties for their services in Conspec format.
- The Marketplace, which acts as an enriched service registry maintaining both the functional and security characteristics of individual service components.
- The Contract Manager Module (CMM), which is part of the Security-by-Contract Component and manages the overall security checking process and checks the compliance of the offered service security level from a service provider with the consumer‘s security policy.
- The Trustworthiness Module (TM), which offers prediction over the trustworthiness value of a service component.
- The Security Property Determination Module (SPDM), which manages the security properties associated with a service.
- The Service Runtime Environment (SRE), which orchestrates the deployment of the composite service process.

Main Outcome

- The specification of the composite service process, enriched with security requirements (consumer policy), expressed in Conspec format.
- A list of service specifications, which satisfy the consumer’s security policy.
- A Web-based implementation of a selected composite service process, which complies with specific security requirements.
- A Boolean response on whether an announcement of an Aniketos compliant service is successful or not (this step might need an additional verification check, as it is described in next Section).

3.3 Security Service Validation and Verification package

Description of the offered functionality

This package offers verification and validation checks to the design, registration and execution of secure services. The service validation process can be invoked, when a composite service has been designed and the service developer needs to check the security characteristics of the involved services within the composition context. The same check can be performed at runtime to validate that the offered security level of the composite service complies with the consumer’s security policy.

Furthermore, this package is used to perform a thorough security check on the properties assigned to a service or the components of a composite service.

Involved Aniketos Components

This package consists of the following Aniketos components:

- The Secure Composition Planner Module (SCPM), which suggests the most secure composite service specifications based on certain security features.
- The Contract Manager Module (CMM), which manages the overall security checking process and checks the compliance of the offered service security level from a service provider with the consumer‘s security policy.
- The Nested Composition Verification Module (NCVM), which verifies the compliance of a service specification with the offered service security level from a service provider.
- The Trustworthiness Module (TM), which offers prediction over the trustworthiness value of a service component.
- The Security Property Determination Module (SPDM), which manages the security properties associated with a service.
- The Composition Security Validation Module (CSVM), which verifies the compliance of a service composition to the offered security properties.
- The Property Verification Module (PVM), which is part of the Verification Component and analyses a service implementation (e.g., based on its source code) for compliance with required security properties (e.g., absence of certain vulnerabilities, enforcement of access control, ensuring data privacy) as expressed in a service contract.
- The Composite Service Security Testing Module (CSSTM), which is part of the Verification Component and is used to detect vulnerabilities in a service specification.
- The Threat Repository Module (TRM), which exposes the list of registered threats and countermeasures encountered in ICT systems.
- The Service Threat Monitoring Module (STMM), which analyses an event referring to a change in the threat level of an offered composite service.
- The Notification Module, which compiles the proper alert and notification messages to be communicated to the application and other involved Aniketos components.

Main Outcome

- A Boolean response that the security properties of a service specification have been verified.
- A list with the security checks performed and the respective result.

3.4 Security Monitoring and Notification package

Description of the offered functionality

This package enables monitoring the execution of composite services and generating alerts when any malfunction in the proper service operation is identified. Such malfunctions can refer to the violation of a service contract and/or the change in the trustworthiness and/or threat level of the offered composite service.

The package enables subscriptions to service monitors for specific types of events. It, then, captures the events produced from the service execution environment and analyses them to generate alerts and notifications at the application layer for potential breach in the experienced secure service provisioning.

Involved Aniketos Components
This package consists of the following Aniketos components:

- The Service Composition Framework (SCF), which is a process modelling tool (based on Activiti Designer) and is used to define rules to an existing composite service process for handling incidents identified during the execution of the service process.
- The Service Runtime Environment (SRE), which orchestrates the subscription to monitors and generates the events during the composite service execution.
- The Service Monitoring Module (SMM), which captures the events generated by the SRE and classifies them according to their type for further use.
- The Service Threat Monitoring Module (STMM), which receives subscriptions of service components to threats and analyses an event referring to a change in the threat level of an offered composite service.
- The Threat Repository Module (TRM), which exposes the list of registered threats and countermeasures encountered in ICT systems
- The Security Policy Monitoring Module (SPMM), which is part of the Security-by-Contract Component and notified of the composite service contract and analyses an event referring to a service contract violation.
- The Security Property Determination Module (SPDM), which manages the security properties associated with a service.
- The Trustworthiness Component (TM), which is notified on the requirement for monitoring the trustworthiness values of the composite service and analyses an event referring to a change in the trustworthiness level of an offered composite service.
- The Notification Module (NM), which receives subscriptions for notifications to specific security events (i.e. contract change, trust level change, security property change, threat level change, etc.) and compiles the proper alert and notification messages to be communicated to the application and other involved Aniketos components.

Main Outcome

- A set of alerts and notification messages, stating the type of malfunction that was identified.
- A set of rules to guide on the proper incident handling at runtime.

4 Conclusion

This chapter presented the Aniketos platform as-a-whole and is intended to be a starting point for anyone interested in its usage and technology behind. The following chapters in this book give more detailed explanations on how the various components work and interact with each other and the environment. Additionally, more than hundred publications from the Aniketos project are available for further detailing.