Chapter 12
The POLIPO Security Framework

Daniel Trivellato, Sandro Etalle, Erik Luit, Nicola Zannone

Abstract Systems of systems are dynamic coalitions of distributed, autonomous and heterogeneous systems that collaborate to achieve a common goal. While offering several advantages in terms of scalability and flexibility, the systems of systems paradigm has a significant impact on systems interoperability and on the security requirements of the collaborating systems. In this chapter we introduce POLIPO, a security framework that protects the information exchanged among the systems in a system of systems, while preserving systems' autonomy and interoperability. Information is protected from unauthorized access and improper modification by combining context-aware access control with trust management. Autonomy and interoperability are enabled by the use of ontology-based services. More precisely, each authority may refer to different ontologies to define the semantics of the terms used in the security policy of the system it governs and to describe domain knowledge and context information. A semantic alignment technique is then employed to map concepts from different ontologies and align the systems' vocabularies. We demonstrate the applicability of our solution with a prototype implementation of the framework for a scenario in the maritime safety and security domain.

Daniel Trivellato
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands, e-mail: d.trivellato@tue.nl

Sandro Etalle
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands, e-mail: s.etalle@tue.nl
Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands, e-mail: sandro.etalle@utwente.nl

Erik Luit
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands, e-mail: e.luit@tue.nl

Nicola Zannone
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands, e-mail: n.zannone@tue.nl

12.1 Introduction

Systems of systems are coalitions of autonomous systems that collaborate to achieve a common goal. The systems in a system of systems often belong to different security domains, which are governed by different authorities employing heterogeneous protocols, vocabularies, data models and organizational structures. Furthermore, systems of systems are often dynamic, with systems joining and leaving the coalition at runtime. An example of system of systems is a fleet of ships from different NATO countries collaborating in a patrolling mission.

Despite offering a high degree of operational flexibility, the systems of systems paradigm has a strong impact on systems interoperability and on the security requirements of the coalition members (hereafter called *parties*). In fact, during the operation of a system of systems parties are required to exchange information (e.g., their current location) with the other members of the coalition. This information, however, might be sensitive and should be accessed exclusively by authorized parties, which may vary depending on the context (e.g., the location of the requester, the criticality of a situation) and the content of the information. Therefore, along with the development of systems of systems comes the demand for a flexible security framework that faces the related security challenges.

In particular, to deal with the dynamic nature of systems of systems, a security framework should incorporate security-relevant contextual information into access control decisions [2]. Contextual information may consist of "basic" environmental conditions (e.g., the location of the requester, the time of access), or more complex conditions derived from the basic ones (e.g., an emergency situation due to the collision between two vessels). Context-aware access control models [2, 6] can be employed to serve this purpose.

In addition, contrarily to centralized systems where users and resources belong to a single, trusted domain, in a system of systems parties often do not know each other beforehand. It is therefore not possible to rely on identity-based approaches to regulate the access to local resources. Trust management [3] has been proposed as a solution to this problem. Trust management is an approach to access control in distributed systems where access decisions are based on the attributes of a requester, which are certified by means of digital credentials. Credentials are certificates issued by a party attesting that a subject has a certain attribute, and they are digitally signed to ensure their authenticity and integrity.

The problem of most of the existing trust management frameworks (e.g., [1, 12, 14]) is that they assume a complete agreement among the parties in a system of systems on the vocabulary used to denote subjects' attributes and to describe the concepts and relationships that characterize a given domain. In dynamic coalitions of heterogeneous systems, however, parties will more likely "speak" different languages and employ different organizational models; nevertheless, they must be able to collaborate to achieve the coalition's goal. As a first step towards enabling mutual understanding and thus interoperability among parties in a system of systems, semantic approaches have been adopted for policy specification [11, 26]. In particular, ontologies have been used to assign a precise structure and semantics to informa-

tion, and to define domain knowledge. Accordingly, parties can refer to ontologies to provide semantics to the terms used to specify their policies and to denote the concepts in the application domain.

The use of ontologies alone, however, is not enough to achieve interoperability. In fact, parties might refer to different ontologies to denote the same (or similar) concepts in the domain. For instance, each NATO country may use different terms (and a different hierarchy) to denote the ranks of the officers on its ships. Semantic alignment techniques [7, 9] need thus to be employed to map concepts from different ontologies, i.e., to align vocabularies and organizational models. A major drawback of the existing semantic alignment techniques is that they require complete knowledge of the ontologies to be aligned. In many systems of systems, however, this requirement is not admissible since parties do not know each other beforehand or might want to keep part of their knowledge base confidential. Therefore, a solution that is effective also when working with partial knowledge needs to be devised.

In this chapter we present POLIPO, a security framework that protects the *confidentiality* and *integrity* of information while enabling *autonomy* and *interoperability* among the parties in a system of systems. POLIPO combines context-aware access control with trust management to protect information from unauthorized access (confidentiality) and improper modification (integrity). Autonomy and interoperability are enabled by the use of ontology-based services. More precisely, parties may refer to different ontologies in the specification of their policies and to describe domain knowledge and context information. This allows each party to employ the organizational model and terminology that they consider most appropriate within their system. The semantic alignment technique presented in [22] is then employed to align their vocabularies, allowing for mutual understanding.

We present an application of POLIPO to a scenario in the maritime safety and security domain, where a prototype implementation of the framework is employed by the parties in the system of systems to protect the local resources. The framework's architecture, inspired by XACML [17], consists of a set of core security components (e.g., the access control and trust management components) complemented with the ontology-based services. All components and services have been implemented following the service oriented architecture paradigm [18] to facilitate their integration and deployment into existing systems of systems. The modularity of the framework allows for the integration of additional services to support the evaluation of policies and provide additional functionalities, such as a reputation system for the identification of trustworthy information sources (see Chapter 13).

The remainder of this chapter is organized as follows. Section 12.2 presents a use case scenario for a system of systems in the maritime safety and security domain, and elicits a set of basic requirements that a security framework for systems of systems should satisfy. Section 12.3 introduces the ingredients of the POLIPO framework and presents the framework's architecture. A prototype implementation of the framework is then presented in Section 12.4. Finally, Section 12.5 concludes the chapter.

12.2 Requirements Elicitation

In this section we first introduce a scenario for systems of systems in the maritime safety and security domain. Then, we identify the main requirements that a security framework for systems of systems should satisfy.

12.2.1 Case Study: Maritime Surveillance

We present a scenario which focuses on the dynamic evolution of systems of systems in response to emergency situations. In particular, we consider a system of systems headed by the European Union (EU) which has the goal of detecting and preventing terrorist attacks against European harbors. The system of systems consists of vessels of the EU Naval Force (EU NAVFOR) which patrol the coast of Somalia, gathering and exchanging information about the vessels transiting in that area. This information is collected, processed and analyzed by an operation control center in Northwood, UK, which coordinates the activities of the EU NAVFOR vessels. In addition to the EU NAVFOR vessels, search and rescue vessels of EU countries may temporarily join the coalition and get access to the information collected by the EU NAVFOR in case of emergency (e.g., to give assistance to vessels getting into troubles). We point out that all the names and events introduced in this scenario are purely fictitious.

The scenario consists of the following steps:

1. A cargo ship called Blue Star is transiting through the Gulf of Aden towards its final destination, Copenhagen. Blue Star is under investigation by the Danish navy because it is suspected of being involved in terrorist activities. The Danish navy has infiltrated agents who are investigating the evolution of these activities.
2. In the proximity of the Dutch coast the cargo ship Blue Star gets into trouble due to a storm and starts drifting. A vessel of the Dutch coastguard (NL-Lifeboat) is nearby and prepares to intervene to give assistance to Blue Star's crew. In order to prepare the intervention, NL-Lifeboat needs information about the cargo transported by Blue Star. By checking the port from which Blue Star departed, NL-Lifeboat infers that the cargo ship has transited off the Somali coast. Therefore, NL-Lifeboat sends a request for additional information about Blue Star also to the EU NAVFOR operation control center in Northwood.
3. Currently, the situation of Blue Star is not considered by the operation control center critical enough to disclose to NL-Lifeboat the intelligence collected by the Danish navy. Therefore, the operation control center does not provide details about the cargo transported by Blue Star to NL-Lifeboat. Furthermore, NL-Lifeboat is requested not to intervene.
4. After a while, however, the situation becomes more critical and the operation control center loses connection with Blue Star. Consequently, the request for intervention from NL-Lifeboat is accepted: NL-Lifeboat is temporarily allowed by

Party	Policy ID	Policy Rule
Operation control center	OCC_1	Operators on vessels of the EU NAVFOR may access all the available information about the ships transiting in the operation area.
	OCC_2	Operators on search and rescue vessels certified by the navy of an EU country may access all the available information about a ship in case that the ship needs assistance.
Dutch navy	NL_1	All search and rescue vessels certified by the Dutch coastguard are certified as search and rescue vessels by the Dutch navy.
Dutch coastguard	CG_1	NL-Lifeboat is a search and rescue lifeboat of the Dutch coastguard.

Table 12.1 Security policies of the parties in the scenario

the operation control center to access the details about Blue Star's cargo. Through this information NL-Lifeboat's operators find out that Blue Star's cargo contains Anthrax that was possibly meant to be distributed to terrorist cells in Europe. The rescuers must use protective clothes and other ships must be kept at a safe distance of at least 500 meters.

Table 12.1 presents the security policies governing the scenario, i.e., the rules defining the authorized accesses to the information controlled by each party in the system of systems. To each policy rule we assign a unique identifier (column *Policy ID*) that we use to refer to the rule in the rest of the paper. Note that the vocabulary used to specify policy rules by the different parties is not always consistent. For example, policy rule NL_1 states the Dutch navy certifies all the "search and rescue vessels" of the Dutch coastguard; NL-Lifeboat, however, is certified as a "search and rescue lifeboat" by the Dutch coastguard (policy CG_1). In this case, an alignment of the vocabularies of the Dutch navy and coastguard is required in order to allow for a certification of NL-Lifeboat by the Dutch navy.

The interactions between the actors in the scenario are shown in Fig. 12.1. Each interaction is labeled with the step of the scenario in which it is described. In case that a step involves multiple interactions, we order them by adding a letter to the label. Since security policies are often deemed to be confidential (see Section 12.2.2 for a more comprehensive discussion), in our scenario we assume that no party has access to the security policies of the other parties. Therefore, each time a party requires a certificate issued by another party, an explicit request needs to be made.

Every time a party receives a request, it evaluates the request against its security policy and returns a response accordingly. Notice, for instance, that the same request for details about Blue Star's cargo sent by NL-Lifeboat to the operation control center in Northwood at two different moments in time (messages 2 and 4a in Fig. 12.1) leads to two different responses (messages 3 and 4f respectively). In fact, since in step 3 of the scenario the operation control center does not consider Blue Star to be needing assistance, policy OCC_2 cannot be applied. On the contrary, in step 4 the situation of Blue Star is considered critical and the details about its cargo are provided to NL-Lifeboat, upon verifying that NL-Lifeboat is a search and rescue vessel certified by the Dutch navy.

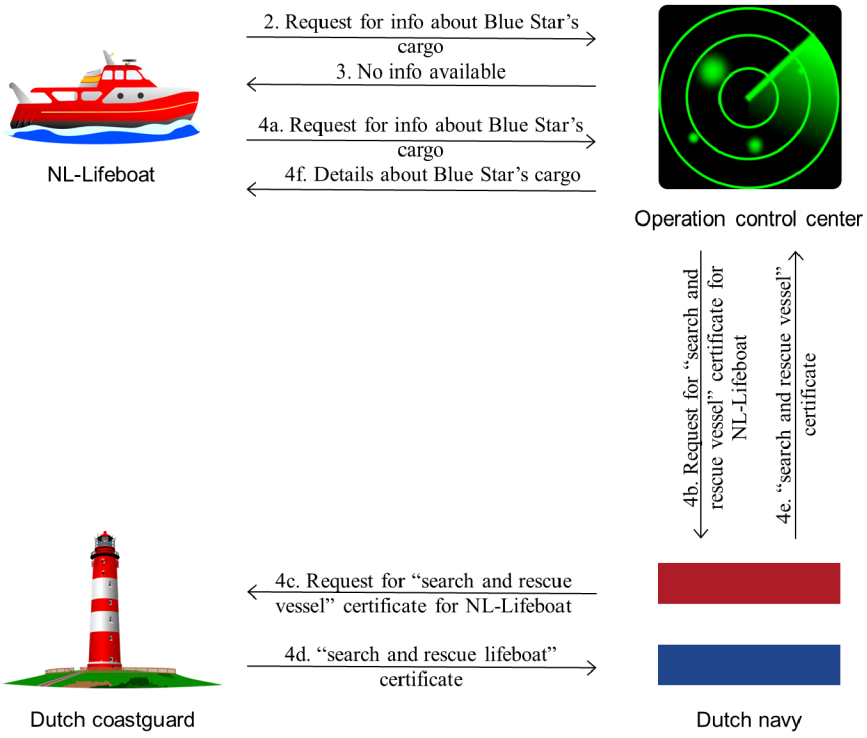


Fig. 12.1 Interactions among the parties in the scenario

12.2.2 Security Requirements

As a first step towards the elicitation of our security objectives, we derive the characteristics of systems of systems which are relevant for the design of a security framework. The distinguishing features of systems of systems are as follows:

- *Dynamicity*: systems of systems are constantly evolving. Systems may leave a system of systems at any time while new systems may join the coalition, depending on the context or the progress towards the goal. In the scenario introduced in the previous section, for example, NL-Lifeboat joins the EU NAVFOR system of systems during the emergency of the cargo ship Blue Star, and leaves the coalition when the emergency is over. Similarly, the information that systems need to exchange may be context-dependent. For instance, in case of emergency parties may be authorized to access sensitive information that they would normally not be allowed to access, such as NL-Lifeboat in our scenario.
- *Distribution*: contrarily to centralized systems where users and resources belong to a single, trusted domain, systems of systems are characterized by the absence of a central point of control. Each system in a system of systems is an independent, complex system which belongs to a (possibly) different security domain

and is governed by a different authority (e.g., the operation control center and the Dutch coastguard). Furthermore, systems of systems are open systems in which parties may not know each other before joining the coalition. For example, the operation control center does not know whether NL-Lifeboat is actually a search and rescue vessel of an EU country, and therefore requests its certification from a trusted party, i.e., the Dutch navy.

- *Heterogeneity*: as a consequence of the autonomy of the systems in a system of systems, each system may adopt different data and organizational structures, and a different vocabulary to define the concepts and relationships in an application domain. In policy CG_1 (Table 12.1), for instance, the Dutch coastguard refers to NL-Lifeboat as a "search and rescue lifeboat", whereas the Dutch navy issues only certificates with attribute "search and rescue vessel" (policy NL_1).

These features impose serious challenges on the design of a security framework. We identify the following set of core security requirements that a security framework for systems of systems should satisfy:

1. *Protection of information confidentiality and integrity*: sensitive data exchanged among the parties in the system of systems must be protected from unauthorized access and improper modification. For example, if terrorists would be able to access the information gathered by the EU NAVFOR, they would know that the Danish navy is investigating the activities of the cargo ship Blue Star. Security policies, however, may also contain sensitive information which needs to be protected. In particular, the disclosure of a security policy may reveal the relationship among some parties in a domain, such as business relationships or alliances, whose disclosure could be exploited by adversary parties, e.g., terrorists [19]. In addition, the disclosure of a policy may leak information that can be used to exploit vulnerable points of a system [20]: by knowing the security policies protecting the intelligence gathered by the EU NAVFOR, for instance, terrorists would know who are the parties that might be aware of their activities, and under which circumstances. Furthermore, by accessing a policy an adversary would know what credentials he needs to forge to illegitimately gain access to a resource [8]. Therefore, the disclosure of both data and policies shall be protected.
   The distributed and dynamic nature of systems of systems introduces two additional challenges to the confidentiality and integrity requirements. More precisely, policies that regulate the access to information need to:
   - take into account that parties may not know each other beforehand. Therefore, authorizations cannot (always) be defined based on the identity of the requester. Rather, they should be based on his attributes (e.g., nationality, rank).
   - be flexible and adaptable to different circumstances. In this respect, the evaluation of access requests should take the current context into consideration (e.g., the criticality of a situation).
2. *Autonomy of the parties involved*: the dynamicity of systems of systems implies that collaborations are often short-lived and the systems involved may change

over time. In addition, the parties in a system of systems are independent systems that have individual objectives next to the ones of the coalition, and may be involved in more than one system of systems at a time. In this setting, we cannot expect the parties in a system of systems to employ common data models, organizational structures, and vocabularies for the specification of their security policies. Rather, parties should be able to employ different models and vocabularies.

3. *Interoperability among parties*: despite the heterogeneity in their organizational structures and vocabularies, parties must be able to understand each other for the success of the coalition. In our scenario, for example, it is evident that NL-Lifeboat should be interpreted by the Dutch navy as a "search and rescue vessel", even though it is defined as "search and rescue lifeboat" by the Dutch coastguard.
4. *Ease of integration into existing systems*: the services and functionalities offered by the systems in a system of systems have strong implications on the design of the security components. On the one hand, the functional components of a system should be designed and implemented as independently as possible from its security components. On the other hand, a security framework for systems of systems should be easy to integrate into existing systems and should easily interface with the system's functionalities to protect the system's confidential information. In addition, the framework should be flexible and allow for an easy integration of additional security components that may become relevant during the lifetime of a system of systems (e.g., a reputation system for service selection).

Clearly, these requirements do not cover all the security aspects that are relevant for systems of systems. For instance, we omit the requirements for the protection of the systems and services in a system of systems from network attacks such as denial of service, eavesdropping, identity spoofing, etc. In the context of the POSEIDON project, however, our focus is mainly on the design of a solution that satisfies the requirements that are characteristic for systems of systems. The combination of our security framework with existing techniques that address other security requirements is out of the scope of this chapter.

12.3 The POLIPO Framework

In this section we present the POLIPO security framework. The contribution of POLIPO lies both in its ingredients, which implement new models and techniques especially designed to meet the security requirements of systems of systems, and in the way in which these ingredients have been combined into a unified framework. In the next two subsections we briefly introduce the framework ingredients (Section 12.3.1) and show how they have been integrated in the POLIPO architecture (Section 12.3.2).

12.3.1 Framework Ingredients

In order to satisfy the requirements introduced in Section 12.2.2, POLIPO combines models and techniques from the fields of computer security, knowledge representation, and software engineering. In particular, it relies on:

1. context-aware access control models and trust management to protect the confidentiality and integrity of information;
2. ontology-based services to enable autonomy and interoperability among the parties in a system of systems;
3. a service oriented architecture to allow for an easy integration and deployment of the framework into existing systems.

In the next paragraphs we will provide more details about each of these techniques.

12.3.1.1 Context-Aware Access Control and Trust Management

Context-aware access control is used to tackle the dynamicity of systems of systems: by incorporating context information (e.g., the location of the requester, the criticality of the situation) in access decisions, parties can specify flexible policies which adapt to different situations. Trust management, on the other hand, deals with the distributed nature of systems of systems. In trust management, access decisions are based on the attributes of a requester (e.g., vessels of the EU NAVFOR), which are certified by means of digital credentials issued by an authority, i.e., any party in the system of systems. The contribution of this approach is twofold: (a) contrarily to identity-based approaches, grounding an access decision on the certified attributes of a requester allows parties to exchange information with (previously) unknown entities; (b) each party can choose which authority to trust for certifying which attributes, and accept only credentials issued by that authority. For example, the operation control center trusts only the navies of EU countries for certifying search and rescue vessels.

To combine context-aware access control with trust management we have defined an ontology-based policy specification language [21]. The language allows for the specification of rules to constrain the access to the local resources of a party (called *authorization rules*) and to define the conditions under which a credential is released (*credential rules*). Context information and domain knowledge (e.g., the type of a ship) are incorporated into authorization and credential rules by referring to a knowledge base, which is represented by a set of ontologies. Examples of authorization rules are rules OCC_1 and OCC_2 in Table 12.1. In rule OCC_2, the need for assistance of a ship is determined by the operation control center by analyzing context information (e.g., whether the ship is drifting) available in the knowledge base. Policy rules NL_1 and CG_1 in Table 12.1 are examples of credential rules. More precisely, in rule NL_1 the Dutch navy states that it is willing to release a "search and rescue vessel" credential only to the vessels in possess of an equivalent credential

issued by the Dutch coastguard. In rule CG_1, the Dutch coastguard certifies that vessel NL-Lifeboat is a "search and rescue lifeboat".

Other existing work [11, 26] proposes the use of ontologies for the specification of security policies. The expressive power of these languages, however, is limited and does not allow for the specification of several types of security constraints, as for instance separation of duty. Since such constraints are common to many application domains for systems of systems (e.g., maritime safety and security, business-to-business), these languages do not provide a valid solution. To overcome this limitation, ontology languages have been extended with rules which enable the specification of more complex constraints [10]. In this respect, our contribution lies in the way in which our rules interface with ontologies: rather than allowing for a free interaction between them, which may lead to the introduction of inconsistencies or cause the ontology reasoning to become undecidable (as in [10]), we only allow information to flow *from* ontologies *to* policy rules. In other words, information from the knowledge base can be used to make informed access or credential release decisions, but policy rules cannot be used to derive new knowledge to be integrated into the knowledge base, as this may make the knowledge base inconsistent due to the introduction of contradicting information.

Furthermore, another contribution is represented by the design and development of a novel algorithm for credential retrieval, called GEM [23]. Contrarily to many of the existing algorithms (e.g., [5, 13]), GEM evaluates requests for credentials in a completely distributed way without disclosing the credential rules of parties, thereby preserving their confidentiality. For example, in the scenario in Section 12.2.1, when requesting the certificate of "search and rescue vessel" for NL-Lifeboat to the Dutch navy, the operation control center cannot infer anything about the certification procedure (i.e., the policy) of the Dutch navy. Similarly, the Dutch navy and the Dutch coastguard are not able to learn each other's policies. As discussed in Section 12.2.2, protecting the confidentiality of security policies is very important, since their disclosure may leak valuable information [19, 20, 8]. Finally, another advantage of GEM is its ability to identify mutually dependent policy rules, preventing loops in the credential retrieval process.

12.3.1.2 Ontology-Based Services

Parties in a system of systems refer to ontologies to assign semantics to the terms used in their policies. More precisely, ontology concepts (or instances) are used to denote the attributes certified by a party's credentials (e.g., "search and rescue vessel"). In addition, ontologies are used to define the data and organizational structures of each party. This, combined with the use of the completely automated semantic alignment technique in [22], which is based on the notion of similarity between ontology concepts, allows parties to use the vocabulary and structures they consider most appropriate within their system (thus accommodating parties' heterogeneity), while preserving mutual understanding with the rest of the coalition.

A major disadvantage of the existing semantic alignment techniques (e.g., [7, 9]) is that they require complete knowledge of the ontologies to be aligned in order to effectively map concepts from different ontologies. When collaborations involve parties that do not know each other beforehand or want to keep part of their knowledge base confidential, however, this condition cannot be guaranteed. To overcome this problem, we adopt the alignment technique introduced in [22] that maps concepts from different ontologies by considering concepts' similarity "estimates" (since they are computed based on partial knowledge) issued by different parties in the system of systems. These estimates are then combined into a single similarity value by weighing them based on the "reliability" of their issuer. In this way, parties can enable interoperability with parties using different vocabularies and increase the flexibility of their policies by accepting credentials about possibly unknown attributes, provided that they are similar to a known attribute for at least a certain degree. For example, in the scenario in Section 12.2.1, the Dutch navy can use the estimates computed by the EU, the operation control center, the Dutch coastguard and itself to assess the similarity between the concepts "search and rescue vessel" and "search and rescue lifeboat". According to the resulting combined similarity estimate, the Dutch navy can then decide whether the two terms are similar enough to certify NL-Lifeboat as a "search and rescue vessel" (policy NL_1 in Table 12.1). The technique in [22] abstracts from the way in which similarity estimates are computed. Several existing algorithms (e.g., [15, 16]) can be employed for this purpose, and each party in the system of systems can adopt a different algorithm without affecting its interoperability with the other parties.

12.3.1.3 Service Oriented Architecture

Service oriented architecture is a paradigm for organizing and utilizing software solutions that promotes reuse and interoperability [18]. A system based on service oriented architecture implements functionality as a suite of interoperable services that can be used within multiple systems from different domains. The characteristic features of service oriented architecture are:

- Service reusability: the logic and functionality of a system is divided into services with the intention of promoting reuse.
- Service autonomy: each service is independent and maintains control over the logic it encapsulates.
- Service loose coupling: the relationship and dependencies among services are minimized.
- Service composability: services can be easily combined and integrated into a larger, complex system.

Service oriented architecture is commonly implemented using web services. Accordingly, our security framework is implemented as a web service which can be easily plugged into existing systems and acts as a proxy server intercepting all the outgoing and incoming messages of a system. Furthermore, each component of the

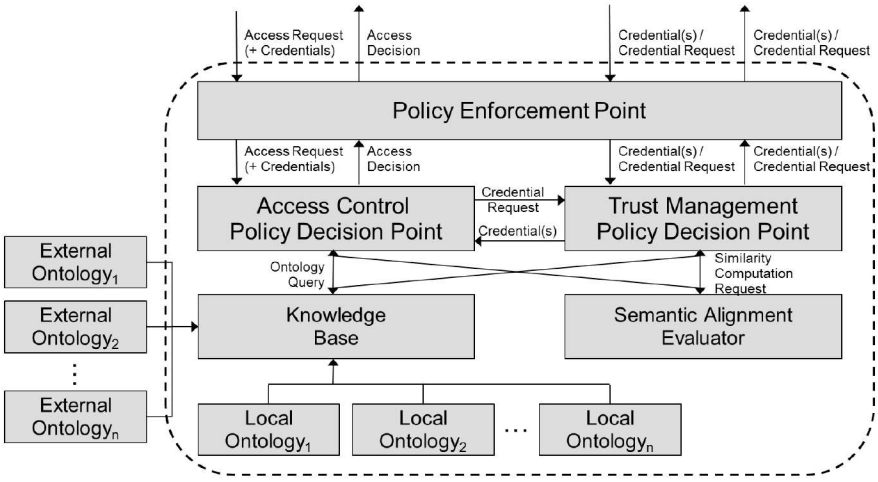


Fig. 12.2 POLIPO framework architecture

security framework introduced in the next subsection is implemented as a service that interfaces with the functional components and the data stored within a system. This modular approach also facilitates the extension of the framework with additional security services that may become relevant during the lifetime of a system of systems (e.g., a reputation system for service selection or a key performance indicator service) [4].

12.3.2 Framework Architecture

Each party in a system of systems can employ an instance of the POLIPO framework to protect the local resources. The POLIPO framework intercepts every access request to a local resource and evaluates it against the local security policy. If the request is authorized by the policy, the resource is returned to the requester; otherwise, the access is denied.

An overview of the framework's architecture is shown in Fig. 12.2, where the dashed line separates the local components from the external ones. POLIPO consists of a set of core components (i.e., *policy enforcement point*, *access control* and *trust management policy decision point*), inspired by the XACML architecture [17], and a number of complementary specialized services used to assist the policy evaluation process (i.e., *knowledge base* and *semantic alignment evaluator*). The combination of the core components with the specialized services ensures confidentiality and integrity of information, while preserving autonomy and interoperability among the parties in a system of systems. In the next paragraphs we discuss these five components and services in detail.

Policy Enforcement Point. The policy enforcement point is the interface of a party with the external world, and has three main tasks: (1) intercepting incoming requests for local resources, (2) contacting the appropriate policy decision point to evaluate those requests, and (3) enforcing the decision of the policy decision point. We consider two types of requests: *access requests* and *credential requests*. Access requests are requests for data controlled by the local party (e.g., message 2 in Fig. 12.1), while credential requests are requests for credentials issued by the local party (e.g., message 4b in Fig. 12.1).

Upon receiving a request, the policy enforcement point forwards it to the appropriate policy decision point. In particular, access requests are processed by the access control policy decision point, while credential requests by the trust management policy decision point. Finally, the policy enforcement point enforces the decision of the policy decision point. If the decision is positive, i.e., if access to the requested data is authorized (in case of an access request) or the requested credential is locally available (in case of a credential request), the policy enforcement point returns the requested resource; otherwise, a "deny" response is sent to the requester.

Access Control Policy Decision Point. The access control policy decision point is responsible for the evaluation of access requests. When it receives an access request, the access control policy decision point checks whether the applicable authorization rules depend on some credentials: if this is the case, they are requested to the trust management policy decision point, which takes over the responsibility of retrieving them. Then, depending on whether all the necessary credentials have been successfully retrieved and the other conditions in the authorization rules (e.g., context conditions) are satisfied, the access control policy decision point determines whether the access request should be authorized or denied. Context information and domain knowledge are retrieved by invoking the knowledge base service, while the alignment between ontology concepts is performed by the semantic alignment evaluator.

Trust Management Policy Decision Point. The trust management policy decision point is responsible for the evaluation of credential requests. The credential retrieval algorithm within the trust management policy decision point defines the procedure to compute the answers to a credential request; in our framework we employ GEM [23] as credential retrieval algorithm. The evaluation of a credential request may depend on credentials which are not locally available and, consequently, need to be retrieved from some other party (e.g., a credential issued by the Dutch coastguard in policy NL_1 in Table 12.1). In this case, the trust management policy decision point sends the request for the missing credential to the policy enforcement point, which forwards it to the appropriate party, and feeds the response back to the trust management policy decision point. As for the access control policy deci-

sion point, ontology queries in credential rules are resolved by the knowledge base service, while ontology mappings are requested to the semantic alignment evaluator.

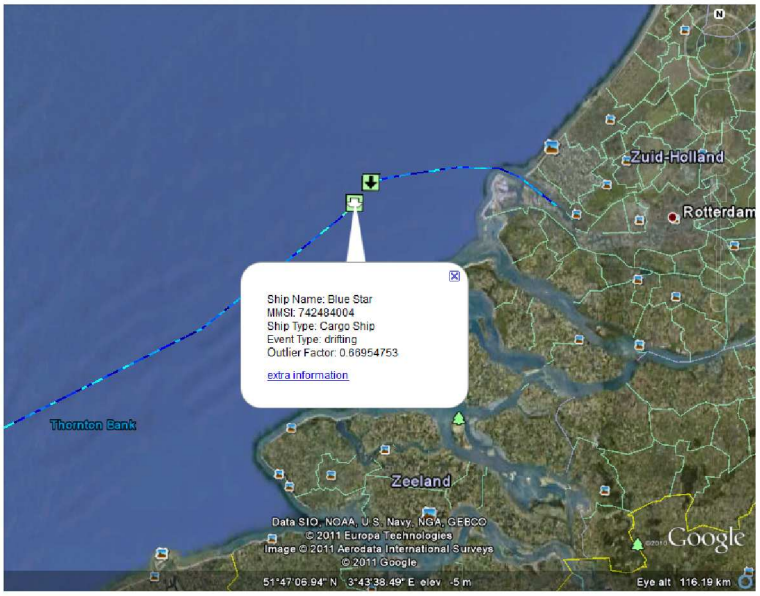
Knowledge Base. The knowledge base service is used by a party to retrieve the context and domain information that is relevant for an access or credential release decision. For example, in rule OCC_2, the need for assistance of a ship is determined by the operation control center based on an "outlier factor" (see Chapter 8) associated to the ship, which is available in the knowledge base. The knowledge base consists of a set of local ontologies that define the concepts and relationship employed in the party's policy, the structure of the information it controls (i.e., metadata annotating the local resources), and all the domain and context information (e.g., the current location of a vessel) collected and derived within the system. A party can enlarge its knowledge base by importing external ontologies defined and published by other parties or institutions. Within the POSEIDON project, POLIPO relies mainly on the simple event model ontology (Chapter 10).

Semantic Alignment Evaluator. This service computes the similarity between ontology concepts, relationships, and instances. The result of the computation is a similarity value which can be used in a constraint in authorization and credential rules. In the evaluation of rule NL_1 in Table 12.1, for instance, the similarity value between "search and rescue vessel" and "search and rescue lifeboat" is used by the Dutch navy to determine whether vessel NL-Lifeboat should be certified as "search and rescue vessel". In Fig. 12.2, the semantic alignment evaluator is depicted as a service directly controlled by the local party; however, parties in a coalition may rely on (possibly shared) external semantic alignment services.

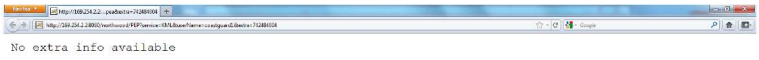
12.4 Prototype Implementation

We have deployed a prototype implementation of POLIPO [24, 25] into a system of systems developed within the POSEIDON project. The system of systems consists of five parties: the operation control center, the Danish navy, the Dutch navy, the Dutch coastguard, and the Dutch coastguard's lifeboat NL-Lifeboat. Parties in the system of systems use Google Earth to visualize and analyze the maritime traffic in their operational area. Communication among parties (i.e., access and credential requests) is via HTTP. In this setting, the policy enforcement point acts as a web proxy that intercepts all the incoming HTTP requests, loads the requested data, and returns an HTTP response based on the policy of the party controlling the data.

In the prototype, the policy enforcement point has been divided into two modules: an interface module and a services module that consists of a set of responders, one for each service offered by the local party. The interface module waits for incoming requests and passes them to the appropriate responder which takes care of processing the request and generating a response. The policy enforcement point of each system in the POSEIDON system of systems has two responders: one to process access requests from the visualization service (Google Earth), and one to process



(a) Data view for the operator of NL-Lifeboat



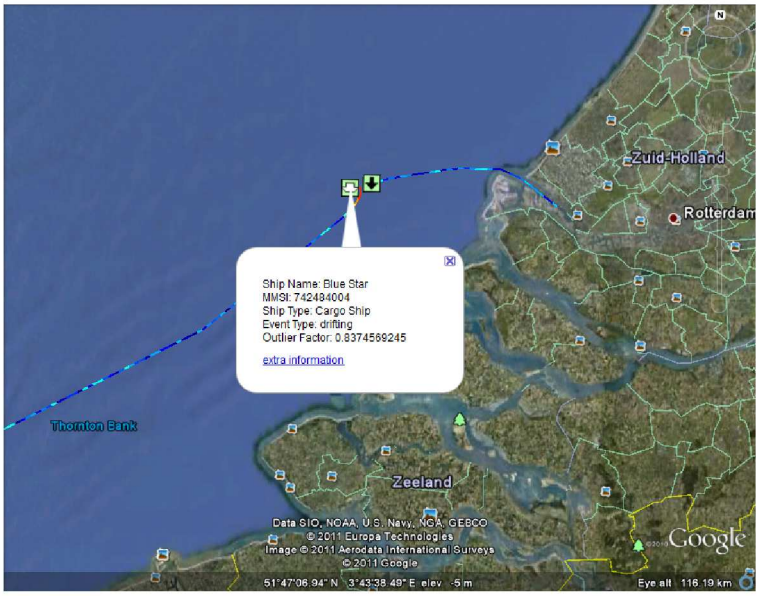
(b) Information about Blue Star's cargo displayed to the operator of NL-Lifeboat

Fig. 12.3 Data view and extra information displayed to the operator of NL-Lifeboat before the emergency

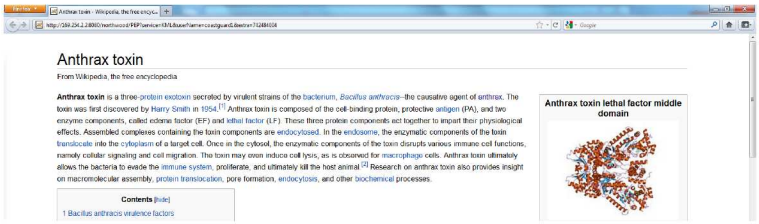
credential requests. The division of the policy enforcement point into two modules enhances the flexibility of the POLIPO framework, as it allows new services offered within the system of systems to be secured by simply adding the relative responders to the policy enforcement point component.

We now show an application of POLIPO to the scenario introduced in Section 12.2.1. Fig. 12.3 and Fig. 12.4 show the output of the visualization service and the details about Blue Star's cargo returned to the operator of NL-Lifeboat respectively before and during the emergency situation declared by the operation control center, based on the policies in Table 12.1. As mentioned in Section 12.2.1, all the names and details about ships introduced in this chapter are purely fictitious.

In the visualization, the color of a segment reflects the outlier factor computed for that segment (see Chapter 8). The color scale goes from blue to red as the outlier factor increases. The current position of each ship is represented by an icon, whose type corresponds to the type of the vessel it represents: more precisely, icons depicting an arrow pointing downwards represent vessels which are (or become) part of the EU NAVFOR; all the other vessel types are represented by an icon depicting



(a) Data view for the operator of NL-Lifeboat



(b) Information about Blue Star's cargo displayed to the operator of NL-Lifeboat

Fig. 12.4 Data view and extra information displayed to the operator of NL-Lifeboat during the emergency

a white boat. By clicking on an icon or a segment the operator can see the name, maritime mobile service identity (MMSI), and type of a ship, as well as the event type (e.g., the ship is slowing down, has stopped, etc.) and the outlier factor associated to the event. For instance, in Fig. 12.3(a) and Fig. 12.4(a), information about the ship Blue Star is displayed. The information box shows that Blue Star is a cargo ship which is currently drifting, and the outlier factor is approximately 0.67 and 0.84 respectively.

When the operator of NL-Lifeboat requests details about Blue Star's cargo to the operation control center, the security framework of the operation control center verifies whether there are rules in the local policy which might grant access to the requested information, and if so, it attempts to collect the credentials of NL-Lifeboat required to authorize the access. For example, the first time that NL-Lifeboat re-

quests the cargo information, the operation control center does not return any details because the situation of Blue Star is not considered critical (Fig. 12.3(b), corresponding to message 3 in Fig. 12.1), and therefore rule OCC_2 in Table 12.1 cannot be applied. On the contrary, when the conditions become more critical (as can be evinced by the high outlier factor associated to Blue Star and the event type - drifting - in Fig. 12.4(a)), the intelligence gathered by the EU NAVFOR about Blue Star's cargo is provided to the operator of NL-Lifeboat (message 4f in Fig. 12.1). In this case, since the cargo contains Anthrax, the information returned to NL-Lifeboat includes details about the Anthrax toxin and measures to prevent the infection (Fig. 12.4(b)). The authorization of NL-Lifeboat to access the intelligence collected by the EU NAVFOR follows from the verification that NL-Lifeboat is a "search and rescue vessel" certified by the Dutch navy. In turn, the certification of NL-Lifeboat as a "search and rescue vessel" from the Dutch navy results from the alignment of the vocabularies of the Dutch navy and the Dutch coastguard (see Table 12.1 for the vocabulary employed in the two parties' policies).

12.5 Discussion and Conclusion

The security challenges in systems of systems are different from those affecting centralized systems. In a dynamic, inter-organizational coalition of systems, parties might not know each other beforehand, might employ different data and organizational models and speak different languages; nevertheless, they must be able to collaborate for the success of the coalition. We have identified four main requirements that a security framework for systems of systems must satisfy: (1) protection of the *confidentiality* and *integrity* of data and security policies; (2) *autonomy* of parties in the choice of data and organizational model and vocabulary used to specify policies and describe the local resources; and (3) *interoperability* among parties. In addition, (4) the security framework must be *easy to integrate* into existing systems.

Several security frameworks for systems of systems have been proposed in the literature. These frameworks can be divided into two categories: semantic frameworks [11, 26] and trust management frameworks [5, 12, 14]. Semantic frameworks rely on ontologies for the specification of security policies and the definition of domain knowledge and context information. This enables interoperability among parties at the cost of limiting the expressive power of the policy language, which does not allow the specification of several types of security constraints (e.g., mutually exclusive roles). On the other hand, trust management frameworks rely on an attribute-based approach to access control where access decisions are based on digital certificates, called credentials. Trust management frameworks employ expressive policy specification languages to ensure data confidentiality and integrity; however, they either assume all parties in a system of systems to use the same vocabulary [12, 14], or do not provide a mechanism to align different vocabularies [5]. Thus, none of the existing frameworks satisfies all the security requirements of systems of systems.

In this chapter we have introduced POLIPO, a security framework for systems of systems satisfying all the aforementioned requirements. Confidentiality and integrity of information are protected by complementing context-aware access control with trust management; a distributed policy evaluation algorithm guarantees that policies' confidentiality is not violated when a credential request is evaluated. Security policies are specified by means of an ontology-based specification language [21]. The use of ontologies allows parties to provide semantics to the terms employed in their policies and to describe domain and context information. This, combined with a semantic alignment technique [22], gives parties the autonomy to employ different organizational models and vocabularies while preserving interoperability among the parties in the system of systems.

The applicability of POLIPO has been demonstrated by a prototype implementation for a scenario in the maritime safety and security domain, where communication among the parties in the system of systems is via HTTP. In this setting, POLIPO acts as a web proxy which intercepts all the incoming access and credential requests directed to a party, and returns a response based on the security policy of the party. This facilitates the deployment of the framework into existing systems. In addition, all the framework components have been implemented following the service oriented architecture paradigm to allow for an easy integration of additional components to support the evaluation of policies and provide additional functionalities.

Even though POLIPO has been mainly tested in systems of systems in the maritime safety and security domain, its characteristics make it suitable for many other domains. For example, we have deployed POLIPO also in systems of systems in the e-health and the employability domain [4]. Furthermore, its integration with ontology-based services allows for an easy deployment of POLIPO into systems of systems on the semantic web. More generally, POLIPO represents a valid security solution for all the domains characterized by the need for collaborations among parties from different security domains, who possibly do not know each other beforehand and employ different vocabularies and different data and organizational structures.

Acknowledgements This research has been carried out as a part of the POSEIDON project at Thales under the responsibilities of the Embedded Systems Institute (ESI). This project is partially supported by the Dutch Ministry of Economic Affairs under the BSIK program.

References

1. Moritz Y. Becker and Peter Sewell. Cassandra: Distributed Access Control Policies with Tunable Expressiveness. In *Proceedings of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY'04*, pages 159–168, Washington, DC, USA, 2004. IEEE Computer Society.
2. Rafae Bhatti, Elisa Bertino, and Arif Ghafoor. A Trust-Based Context-Aware Access Control Model for Web-Services. *Distributed and Parallel Databases*, 18(1):83–105, July 2005.

3. Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized Trust Management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy, SP'96*, pages 164–173. IEEE Computer Society, 1996.
4. K. Böhm, S. Etalle, J. den Hartog, C. Hütter, S. Trabelsi, D. Trivellato, and N. Zannone. Flexible Architecture for Privacy-Aware Trust Management. *Journal of Theoretical and Applied Electronic Commerce Research*, 5(2):77–96, August 2010.
5. Marcin Czenko and Sandro Etalle. Core TuLiP logic programming for trust management. In *Proceedings of the 23rd international conference on Logic programming, ICLP'07*, volume 4670 of *Lecture Notes in Computer Science*, pages 380–394, Porto, Portugal, 2007. Springer-Verlag.
6. Anand Dersingh, Ramiro Liscano, and Allan Jost. Context-aware access control using semantic policies. *Ubiquitous Computing And Communication Journal (UBICC) - Special Issue on Autonomic Computing Systems and Applications*, 3:19–32, 2008.
7. AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Halevy. Learning to match ontologies on the Semantic Web. *The VLDB Journal*, 12(4):303–319, November 2003.
8. Keith Frikken, Mikhail Atallah, and Jiangtao Li. Attribute-Based Access Control with Hidden Policies and Hidden Credentials. *IEEE Transactions on Computers*, 55:1259–1270, October 2006.
9. Steven Heeps, Joe Sventek, Naranker Dulay, Alberto Schaeffer Filho, Emil Lupu, Morris Sloman, and Stephen Strowes. Dynamic Ontology Mapping for Interacting Autonomous Systems. In *Proceedings of the 2nd International Workshop on Self-Organizing Systems, IWSOS'07*, volume 4725 of *Lecture Notes in Computer Science*, pages 255–263, The Lake District, UK, 2007. Springer.
10. Ian Horrocks, Peter F. Patel-Schneider, Sean Bechhofer, and Dmitry Tsarkov. OWL rules: A proposal and prototype implementation. *Journal of Web Semantics*, 3(1):23–40, 2005.
11. Lalana Kagal, Massimo Paolucci, Naveen Srinivasan, Grit Denker, Tim Finin, and Katia Sycara. Authorization and Privacy for Semantic Web Services. *IEEE Intelligent Systems*, 19(4):50–56, 2004.
12. Ninghui Li, John C. Mitchell, and William H. Winsborough. Design of a Role-Based Trust-Management Framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy, SP'02*, pages 114–130, Washington, DC, USA, 2002. IEEE Computer Society.
13. Ninghui Li, William H. Winsborough, and John C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1):35–86, February 2003.
14. Wolfgang Nejdl, Daniel Olmedilla, and Marianne Winslett. PeerTrust: Automated Trust Negotiation for Peers on the Semantic Web. In *Proceedings of the 2004 VLDB Workshop on Secure Data Management, SDM'04*, volume 3178 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2004.
15. Le Duy Ngan, Tran Minh Hang, and A. E. S. Goh. Semantic Similarity between Concepts from Different OWL Ontologies. In *Proceedings of the 5th IEEE International Conference on Industrial Informatics, INDIN'06*, pages 618–623, Los Alamitos, CA, USA, 2006. IEEE Computer Society.
16. Hoa A. Nguyen and Hisham Al-Mubaid. A Combination-based Semantic Similarity Measure using Multiple Information Sources. In *Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration, IRI'06*, pages 617–621. IEEE Systems, Man, and Cybernetics Society, 2006.
17. OASIS. eXtensible Access Control Markup Language (XACML) Version 2.0. Technical report, OASIS Standard, 2005. `http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf`.
18. OASIS. Reference Model for Service Oriented Architecture 1.0. OASIS Standard, OASIS Standard, 2006. `http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf`.
19. Kent E. Seamons, Marianne Winslett, and Ting Yu. Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation. In *Proceedings of the Network and Distributed System Security Symposium, NDSS'01*, San Diego, CA, USA, 2001. The Internet Society.

20. Kevin Stine, Rich Kissel, William C. Barker, Annabelle Lee, and Jim Fahlsing. Guide for Mapping Types of Information and Information Systems to Security Categories. Special Publication SP 800-60 Rev. 1, National Institute of Standards and Technology (NIST), 2008.
21. D. Trivellato, F. Spiessens, N. Zannone, and S. Etalle. POLIPO: Policies & OntoLogies for Interoperability, Portability, and Autonomy. In *10th IEEE Int. Symp. on Policies for Distributed Systems and Networks (POLICY'09)*, pages 110–113. IEEE Computer Society, 2009.
22. D. Trivellato, F. Spiessens, N. Zannone, and S. Etalle. Reputation-Based Ontology Alignment for Autonomy and Interoperability in Distributed Access Control. In *IEEE Int. Conf. on Computational Science and Engineering (CSE'09)*, volume 3, pages 252–258. IEEE Computer Society, 2009.
23. D. Trivellato, N. Zannone, and S. Etalle. GEM: A Distributed Goal Evaluation Algorithm for Trust Management. Computer Science Report CS 10-15, Eindhoven University of Technology, 2010. `http://alexandria.tue.nl/repository/books/695281.pdf`.
24. D. Trivellato, N. Zannone, and S. Etalle. A Security Framework for Systems of Systems. In *12th IEEE Int. Conf. on Policies for distributed systems and networks (POLICY'11)*, pages 182–183. IEEE Computer Society, 2011.
25. D. Trivellato, N. Zannone, and S. Etalle. Poster: Protecting Information in Systems of Systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, *18th ACM Conf. on Computer and Communications Security (CCS'11)*, pages 865–868. ACM, 2011.
26. Andrzej Uszok, Jeffrey M. Bradshaw, Matthew Johnson, Renia Jeffers, Austin Tate, Jeff Dalton, and Stuart Aitken. KAoS Policy Management for Semantic Web Services. *IEEE Intelligent Systems*, 19(4):32–41, 2004.