Self-Healing and Secure Adaptive Messaging Middleware for Business-Critical Systems

Habtamu Abie
Norwegian Computing Center,
Oslo, Norway
e-mail: Habtamu.Abie@nr.no

Reijo M. Savola
VTT Technical Research Centre of Finland,
Oulu, Finland
e-mail: Reijo.Savola@vtt.fi

John Bigham
Queen Mary, University of London, London, UK
e-mail: john.bigham@elec.qmul.ac.uk

Ilesh Dattani
Q-Sphere Ltd, London, UK
e-mail: ilesh@q-sphere.com

Domenico Rotondi
TXT e-solutions SpA, Valenzano (BA), Italy
e-mail: Domenico.Rotondi@TXTGroup.Com

Giorgio Da Bormida
CNIT, Florence, Italy
e-mail: dabormida@gmail.com

Abstract — **Current business-critical systems have stringent requirements for the significant and measurable increase in the end-to-end intelligence, security, scalability, self-adaptation and resilience. Existing state-of-the-art messaging systems achieve arbitrary resilience by a brute-force approach. Self-healing is either rudimentary or non-existent. In this study we present a self-healing and secure adaptive messaging middleware that provides solutions to overcome limitations in robustness, resilience, self-adaptability, scalability, and assurance against security threats and erroneous input during run-time in the face of changing threats. This developed system supports a messaging infrastructure which enables adaptive functions and assurance against security vulnerabilities and erroneous input vulnerabilities to improve the reliability, robustness and dependability of business-critical infrastructures. It provides autonomous adjustments of the run-time configuration of the system in order to preserve and maintain optimal and uninterrupted operation, improvement of the strength of security and degree of trust in the system, and improvement of the assessability and verifiability of the trustworthiness of the system. The methodology used in this research is partly analytical and partly experimental. We develop the new core functionalities theoretically and validate them practically by prototyping.**

Keywords - Self-Adaptation, Messaging Middleware, Self-healing, Resilience, Self-protection, Adaptive Security, Security Metrics

I. INTRODUCTION

The environment surrounding modern business-critical systems is in a continuous state of change throughout the lifetime of an application. With the increase in the dependence of businesses on messaging middleware systems, the need for dependable, trustable, robust and secure adaptive messaging systems becomes ever more acute.

The primary contribution of this work is the analysis and synthesis of a self-healing and secure adaptive messaging middleware for business-critical systems introduced in our earlier work [1]. This paper analyzes (i) the autonomous adjustments of the run-time configuration of the system the purpose of which is the preservation and maintenance of optimal and uninterrupted operation, (ii) the improvement of the strength of security and degree of trust in the system, (iii) the improvement of the assessability and verifiability of the trustworthiness of the system, and (iv) the adaptive integration of the GEMOM solution that consists of a continuous cycle of monitoring, measurement, assessment, optimization, self-healing, adaptation and evolution to meet the challenges in the changing environments.

Message-Oriented Middleware (MOM) provides the functionality of interoperability, portability, and flexibility of architectures that enables applications to exchange messages with other applications without having to know what platform the other application resides on [2][3][4]. MOMs provide a service that allows content providers and consumers to concentrate on the production and consumption of transmitted information. In essence, MOM is compatible with, and can be viewed as, a central component of the Enterprise Service Bus (ESB) [5] architecture, where the MOM message broker acts as the 'bus' between applications. The term Publish/Subscribe (PS) MOM does not necessarily imply the broad collection of concepts and standards of ESB. The key advantage of the MOM architecture is that it reduces the number of point-to-point connections in a complex business-critical IT system.

However, existing commercial MOM technologies are expensive and lack scalability. In addition, there are no solutions that provide the required levels of robustness, reliability and resilience appropriate for future real-time and business-critical systems. Moreover, because they must be self-organizing, modern autonomous MOM platforms have stringent requirements for resilience (ability to keep going in given scenarios by learning, evolving, etc., over time), self-healing (ability of the system to preserve its capabilities even in the event of failure of any individual or multiple components), self-learning and self-optimization, self-adaptation and evolution, fault-tolerance, self-active-vulnerability assessment, adaptive autonomic security. GEMOM (Genetic Message-Oriented Secure Middleware) provides solutions to overcome these limitations to secure messaging to support a communications framework that can be deployed for a wide range of applications [6][7].

Complex, distributed business-critical systems are virtually impossible to implement without the heavy use of a messaging infrastructure. The most common variant of these systems is the scheme utilizing the PS messaging paradigm. Synchronous Request/Reply is easily overlaid on top of PS, making PS the right proxy for overall messaging. GEMOM uses the PS messaging paradigm and further supports better interoperability and integration of business-critical systems by allowing actual instances to be configured so various functions are subcontracted to one or more separate, external or federated entities. GEMOM [7][1] has made advances in the following areas: resilience and self-healing, scalability and resilience, integrated vulnerability management, better interoperability and integration of distributed business-critical systems, and holistic and systematic adaptive security monitoring and measurement.

The rest of this paper is organized as follows. Section II gives an overview of related work. Section III presents the GEMOM system architecture with a brief overview of key properties of the system. In Section IV, we describe the self-optimization, self-healing and scalability of GEMOM. The holistic and systematic adaptive security approach is presented in Section V. In Section VI, we describe the enhanced interoperability and integration of distributed business-critical systems. Section VII shows how the self-healing, adaptive security and the different tool-sets are integrated. A brief introduction to GEMOM prototypes and validation is given in Section VIII. The paper closes with a conclusion and future work in Section IX.

II. Related Work

This section gives a rundown of related work and comparisons of our work with that of whose work is most closely related to ours.

A. MOM Systems

MOM platforms are available in a wide range of implementations such as JMS, WebSphereMQ, TIBCO, Herald, Hermes, SIENA, Gryphon, JEDI and REBECCA where each of these MOMs has been designed to achieve specific goals, and employs unique functionality to meet specific messaging challenges [8]. However, the current state-of-the-art technologies do not allow security mechanisms to actually predict or anticipate future threats, and to adapt to rapidly changing behaviours and threats over time. Table I describes the key functionalities of MOMs, the limitations of existing MOM Systems, and the GEMOM advances as comparison as shown in the table below.

TABLE I. Contrasting MOM Functionalities

Key functionalities	Limitations of existing systems	GEMOM advances
Performance - throughput & latency. (*Throughput* represents the number of requests served by the MOM per second). *Latency* is the time between publishing a message and the subscriber receiving it	Insufficient information and control over performance	Externalised architecture for monitoring resilience, which limits impedance of processing rates while offering control
Increasing interoperability, portability, and flexibility of architectures	Data-loss prone - no means to compensate for the reliability loss and to integrate limitation of risk of loss into the system offered to the user	Compensation for the reliability loss by automatically finding another source of redundancy
Publish/subscribe – asynchronous	No system re-factoring at runtime	Hot standby brokers with instant switch-over and no data loss
Resilience, self-healing and scalability	Prone to feed failures, arbitrary resilience by a brute-force approach, self-healing is either rudimentary or non-existent, and risk is not quantified	Integration of the tool-sets for the management of threats and of vulnerabilities, intelligent techniques to support security assurance, clustering of namespaces/ topics into namespace with namespace replication, and quantification of risk
Security management	No holistic or systematic adaptive security approach	Adaptive security management based on security evidence information offered by security metrics

B. Self-Healing and Self-Adaptation

Self-healing systems attempt to 'heal' themselves in the sense that they recover from faults and regain normative performance levels by employing models, whether external or internal, to monitor system behaviour and by using inputs to adapt themselves to the run-time environment [9]. Self-adaptive systems aim at anticipating changes which occur in a complex environment and automatically dealing with them at run-time, on the basis of the knowledge of what is happening in the system, guided by objectives and needs of stakeholders [10]. Self-adaptive software evaluates its own behaviour and changes it when the evaluation indicates that the software is not accomplishing what it is intended to do, or when this will lead to better functionality or performance [11]. Self-adaptive systems are characterized by three core functionalities: monitoring (sensing) the environment to recognize problems, making decisions on which behaviour to exhibit, and realizing the behaviour change by adaptation [10][11][12].

A number of surveys of mechanisms and techniques to achieve self-healing and self-adaptation exist. Kramer [13] gives a survey of self-adaptive parameter control in evolutionary computation, classifies self-adaptation in the taxonomy of parameter-setting techniques, gives an overview of automatic online-controllable evolutionary operators, and provides a coherent view of search techniques in the space of strategy parameters, and concludes that self-adaptation is an efficient way to control the strategic parameters of an evolutionary optimization algorithm automatically during optimization. In [14], a classification of adaptation on the basis of the mechanisms used and the level at which adaptation operates within the evolutionary algorithm has been developed. Their classification covers all forms of adaptation in evolutionary computation. Ghosh, Sharman, Rao, and Upadhyaya present a survey and synthesis of self-healing systems and propose a strategy of synthesis and classification [9]. Miorandi, Yamamoto, and Pellegrini present a survey of evolutionary and embryogenic approaches to autonomic networking, applicable to network-level functionalities [15]. They give an overview of the major technical challenges to be met by anyone applying the surveyed techniques to autonomic systems.

A number of self-healing and self-adaptive systems have been recently developed supporting healing and adaptation at different levels. Rodero-Merino, Fernandez, Lopez, and Cholvi propose a topology self-adaptation mechanism for efficient resource location that makes the network change its topology to maintain an efficient configuration that depends on the system load and the peer's capacities [16]. Dustdar, Goeschka, Truong, and Zdun have also proposed self-adaptation techniques for complex service-oriented systems, which comprise model-driven compliance support, runtime interaction mining, run-time management of requirements, and explicit control-loop architecture [17]. Alencar and Weigand [18] present the challenges involved in the predictive self-adaptation of service bundles in a service-oriented scenario in terms of time and cost involved in the adaptation that can be useful to enhance the decision-making process in a business strategic or tactical context. Gjørven, Rouvoy, and Eliassen describe a technology-agnostic self-adaptation middleware for service-oriented architectures that can support a cross-layer adaptation of SOA systems and they show that their middleware is able to exploit both the technologies of the service interface and application layers to support a coordinated adaptation of both layers [19]. Reinecke, Wolter, and Moorsel [20] propose a framework and methodology for the definition of benefit-based adaptivity metrics that allow an informed choice between systems based on their adaptivity to be made, and provide a broad survey of related approaches that may be used in the study of adaptivity and to evaluate their respective merits in relation to the proposed adaptivity metric. Giannakopoulos and Palpanas [21] propose an adaptive subscription service architecture, concerning the update of the clients of an entity name system with information on entity changes, using information from user feedback to model user needs, taking into account both the type and the content of changes.

Our self-healing and secure adaptive messaging middleware is inspired by the work above but is focussed more on providing resilience, self-healing, scalability, integrated vulnerability management, better interoperability and integration of distributed business-critical systems, and holistic and systematic adaptive security monitoring and measurement.

C. Adaptive Security

There have been a number of adaptive security systems that have been developed recently supporting adaptation at different levels and for a number of reasons. Chess, Palmer, and White outline a number of security and privacy challenges facing those designing and developing autonomic systems, and also a number of ways that autonomic principles can be used to make systems more secure than they are today [12]. Hager [22] has in his dissertation developed a context-aware and adaptive security for wireless networks, with application to a pervasive networking environment. Shnitko describes an approach to the design of complex secure systems based on the formalization of adaptive functions in an information-security context, and both practical and theoretical aspects related to the usage of adaptive security in complex systems [23]. Son, Zimmerman, and Hansson propose an adaptable security manager for real-time transactions featuring adaptability and multi-level security services that can be applied in a soft real-time environment in order to achieve performance gains [24]. Schneck and Schwan [25] present an adaptive authentication for networked applications with a novel security control abstraction with which trade-offs in security versus performance may be made explicit.

Zou, Lu, and Jin [26] present an architecture and fuzzy adaptive security algorithm in an intelligent firewall where a fuzzy controller is the core module and the characteristics of packets are fuzzified as its inputs. Abie, Spilling, and Foyn [27] and Abie [28] propose self-contained objects for secure information-distribution systems that carry with them usage rights and enforce on their own behalf these rights assigned to them, preserving their confidentiality and integrity. Pietzowski, Satzger, Trumler, and Ungerer propose a bio-inspired self-protecting organic message-oriented middleware with artificial antibodies that evaluates optimal parameter-setting techniques to minimize the memory space needed for storing the antibodies and to reduce the time needed for detecting malicious messages [29]. Djordjevic, Nair, and Dimitrakos present a virtualized trusted computing platform for adaptive security enforcement of web-services interactions by providing virtual machine-level separation that maps from logical domains imposed by web-service-level enforcement policies [30]. Luo, Ni, and Yong [31] and Ma, Abie, Skramstad, and Nygaard [32] propose trustworthiness assessment methods for the calculation of the degree of trust in a grid computing environment and digital records management over time, respectively. Boukerche and Ren present a trust-based security system for ubiquitous and pervasive computing environments, a trust model that assigns credentials to nodes, updates private keys, manages the trust value of each node, and makes appropriate decisions about nodes' access rights [33]. Goovaerts, Win, and Joosen present a bus-based architecture for integrating security middleware services for achieving flexible and adaptive security middleware with a qualitative comparison of the flexibility of the approach with an alternative aspect-oriented-middleware-based approach [34].

A survey of approaches to adaptive application security, and adaptive middleware can be found in [35] and [36], respectively. A taxonomy of compositional adaptation and a comparison of two approaches for achieving flexible and adaptive security middleware can also be found in [37] and [34], respectively. Presentations of semantic and logical foundations of an adaptive security infrastructure can be found in [38].

It was the work of, inter alia, the above researchers that convinced us of the viability of adaptive security, and therefore gave us confidence in the productivity of our research in this direction. Table II gives a brief comparison of our adaptive security work with other closely related work with their special features and benefits categorized according to their types of adaptation.

TABLE II. BRIEF SURVEY OF ADAPTIVE SECURITY AND TRUST

Adaptation type	References	Features and benefits	Limitations	Advances in our approach
Risk	McGraw [39]	Risk-adaptable access control that bases its access decision on a computation of security risk and operational need	Lack of assessability and verifiability of the trustworthiness of the system	Combination of trust-based security and security-based trust. The integration of a continuous cycle of monitoring, assessment and evaluation, and tools and processes for pre-emptive vulnerability testing and updating.
	Qu and Hariri [40]	Anomaly-based self-protection against network attacks	Lack of models for trust and policy adaptation	Combination of trust-based security and security-based trust
Trust	Ryutov, Zhou, Neuman, Leithead, and Seamons [41]	Adaptive trust negotiation and access control for flexible policy adaptation and capturing dynamically changing system security requirements using user and system suspicion levels.	Lack of integration of a continuous cycle of monitoring, assessment and evaluation models.	Integration of a continuous cycle of monitoring, assessment and evaluation, and tools and processes for pre-emptive vulnerability testing and updating,
	Shrobe, Doyle, and Szolovits [42]	An active trust management for autonomous adaptive survivable systems for compromise-based trust management model	Lack of close integration with adaptive security to minimizing the rate and severity of compromises	Combines risk-based security and a security-based trust model using an adaptive control loop for the provision of a secure communication environment.
Security	Djordjevic, Nair, and Dimitrakos [30]	Trusted computing for security enforcement of web services	Lack of models for integration of assessability and verifiability models	Integration of a continuous cycle of monitoring, assessment and evaluation, and tools and processes for pre-emptive vulnerability testing and updating,
	Weise [43]	A security architecture and adaptive security which is capable of reducing threats and anticipating threats before they are manifested, and uses biological and eco-system metaphors	Lack of models for trust building and for integration of assessability and verifiability models	Combines a compromise-based trust model to maximize the value of risk-taking, and integrates a continuous cycle of monitoring, assessment and evolution, and tools and processes for pre-emptive vulnerability testing and updating.
Policy	Venkatesan and Bhattacharya [44]	Threat-adaptive security policy that adapts security policies according to threats	Lack of trust model to maximize the value of risk-taking, and integration of assessability and verifiability models	Combines a compromise-based trust model to maximizing the value of risk-taking and integrates a continuous cycle of monitoring, assessment and evolution, and tools and processes for pre-emptive vulnerability testing and updating.
	Lamanna [45]	Adaptive security policies enforced by software dynamic translation	Lack of trust model to maximize the value of risk-taking and to minimize the rate and severity of compromises	Combines a compromise-based trust model to maximize the value of risk-taking and integrates a continuous cycle of monitoring, assessment and evolution, and tools and processes for pre-emptive vulnerability testing and updating.

D. *Adaptive Security Metrics*

Security metrics provide the on-line means with which to score different security solutions in adaptive security management. In addition, metrics can be used off-line for security engineering decision-making during the whole lifecycle of the system.

The security metrics development approaches that are most valuable in adaptive security management, focus on security-enforcing mechanisms and the quality of the overall security of the system, are briefly discussed here. Wang and Wulf describe their general-level security metrics development framework based on a decomposition approach in [46]. Heyman, Scandariato, Huygens, and Joosen [47] use a security objective decomposition approach and associate the metrics with security patterns. Savola and Abie apply Wang and Wulf's approach to security requirements and enhance it by a complete description of their entire methodology from the analysis of threats and vulnerabilities to a balanced and detailed collection of metrics, and present an initial collection of security metrics for GEMOM in [48]. The Common Vulnerability Scoring System (CVSS) [49] is an initiative aiming at providing an open and standardized method for rating vulnerabilities. The CVSS, along with some other security vulnerability and weakness metrics systems, has been integrated by the U.S. National Institute of Standards and Technology (NIST) into Security Content Automation Protocol (SCAP) [50]. The purpose of this effort is to develop solutions that will be widely-accepted, but it is not complete; it lacks means to obtain evidence of the security level of security-enforcing mechanisms and methodologies to relate the metrics to security objectives. Howard, Pincus, and Wing [51] and Manadhata, Kaynar, and Wing [52] propose an abstract *attack surface* measurement method. Attack surface means the parts that can be accessed by unauthenticated users, such as attackers, including the set of entry points, exit points, the set of channels and the set of non-trusted data items. Further surveys of security metrics can be found in [53][54][55][56].

III. GEMOM System Architecture and Key Properties

GEMOM exploits the predominant PS [2][57] variant of MOM. For completeness, it provides a synchronous Request/Reply overlay as well. In GEMOM, publishers of messages do not send their messages directly to specific receivers. The published messages are positioned in a hierarchy of logical channels (called namespaces and topics in GEMOM) without the publishers having explicit knowledge of what subscribers there may be. Publishers are loosely coupled to subscribers and need not even know of their existence. Namespaces are a hierarchical classification of topics.

The GEMOM system achieves a considerable increase in the end-to-end resilience of complex distributed business-critical systems to ensure secure transmission of data and services across heterogeneous infrastructures and networks. The GEMOM platform consists of the following resilience and self-adaptive properties: (i) reliability of message sourcing and delivery, (ii) scalability in messaging, (iii) replication of structural and dynamic properties of security policies with adaptive authentication and authorization model, (iv) process-zoning and overall encapsulation to an arbitrary level, and (v) new techniques and tools for pre-emptive and automated checking a deployed system for robustness and vulnerabilities to faults, oversights and attacks, all of which are described in detail in the ensuing sections. In the following subsections, we briefly present the system architecture and key properties.

A. *GEMOM System Architecture*

The GEMOM [7] system architecture is composed of a set of communicating nodes, G-Nodes. Some of these G-Nodes are operational (micro) nodes and some managerial (macro) nodes, see Figure 1. The operational G-Nodes can be classified as Message Brokers (Bs), Clients (either publishing or subscribing messages, Publishers (Ps) or Subscribers (Ss)), Authentication and Authorization Modules (AAMs), Anomaly Detector (AD), Security Measurement Module (SMM), etc. They communicate with managerial nodes of different types. The managerial G-Nodes can be classified as Adaptive Security Managers (ASMs), Audit and Logging Modules (ALMs), and Security Monitoring Tools (SMTs) with associated Security Monitors (SMs) and Quality of Service (QoS) Monitors (QMs), Resilience Managers (RMs), Security Anomaly Managers (SAMs), etc. The managerial G-Nodes make decisions about the run-time operation of the system and require a wider perspective than the individual operational G-Nodes. In GEMOM, a Message Broker is a package consisting of an application server, numerous plug-and-play objects, configuration files, and database schemas.

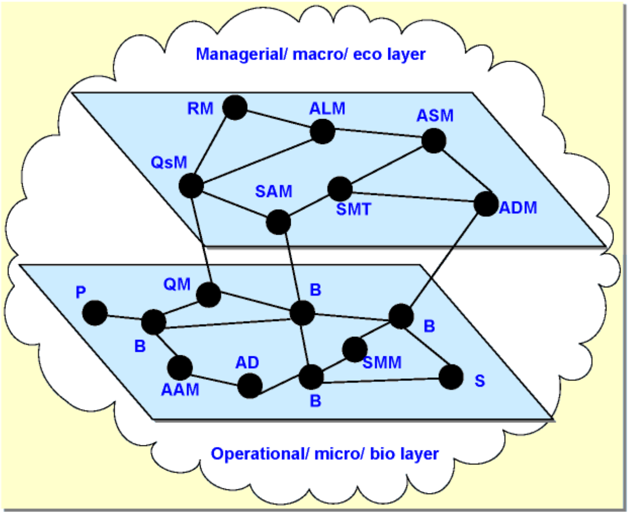

Figure 1. GEMOM system architecture [1]

Figure 1 depicts the GEMOM system architecture showing the main components. It supports mechanisms for adding G-Nodes, for measuring security and QoS between overlay components and publishers and subscribers and deciding what action is to be taken to mitigate loss of security or QoS, or breakdowns, for discovering and communicating with other components in the overlay network, for evaluating the performance of the system in the context of the monitored performance, for establishing the state of the overlay network, and for making decisions on the reconfiguration of routing and message-passing. It also learns from experience and uses its new knowledge in its prediction and decision-making [58].

The biological and ecosystem metaphors provide interesting parallels to the conceptualizations and descriptions of the G-Nodes. The overall GEMOM system architecture has a structure similar to that of a complex adaptive system that utilizes autonomic systems mimicking biological auto-immune systems at the microscopic level (operational level in this case) and that utilize the behaviours of an ecosystem of disparate entities at the macroscopic level (managerial level in this case). Biological and ecological systems maintain system integrity by reacting to foreseeable changes, adapting to unforeseeable changes, or dying. The adaptations and responses can be at a macroscopic ecosystem level (e.g., system or species) or a microscopic biological level (e.g., molecular, cellular) [43]. Hence we can consider GEMOM as having a genetic makeup [7][59].

B. Reliability of Message Sourcing and Delivery

GEMOM supports redundant message feeds (topics and namespaces) and redundant delivery paths (message communication architecture). In the event of failure, switch-over to a redundant resource would be transparent to the end users, with no information loss. As well as entire Message Broker redundancy, GEMOM offers the redundancy of certain subsets or messaging segments. As part of its self-healing functionality when a backup resource is mobilised to carry messages, other nodes, feeds or paths are identified as mirrors (backups) in case of further failure in order to maintain the same level of resilience. This ensures that there are no single points of failure even as new nodes become compromised and so rendered alien and isolated, or even as their rights are revoked.

C. Replicating Structural and Dynamic Properties

One particular GEMOM setup might be configured with a certain security layout or profile in place. GEMOM ensures that the security profiles of the overall system and individual message paths and dynamics are not compromised as a result of failovers. Namely, GEMOM is capable of fully replicating structural and dynamic properties of security policies representing different security layouts or profiles.

GEMOM utilizes a novel Adaptive Security and QoS model that consists of a continuous cycle of monitoring, assessment, and evolution to meet the challenges of the changing environments and threats [7]. This involves gathering contextual information both within the system and the environment, analyzing the collected information and responding to changes by adjusting security functions such as selecting suitable encryption schemes, security protocols, security policies, security algorithms, different authentication and authorization mechanisms, etc. Information gathering for adapting is implemented by using anomaly detection and security monitoring services that register external influences of the environment [60].

The GEMOM project has investigated a number of possibilities in connection with the self-learning capabilities and optimization approaches with respect to resilience. In that sense, the algorithmic approaches for this involved the use of genetic and evolutionary techniques at some level as partial elements for the overall solution.

D. Notion of Faults in GEMOM

The term ‘fault’ in GEMOM refers to a very general concept covering network faults, congestion, and security vulnerabilities, etc. Faults can manifest themselves in the deterioration of the functional profile of the informational system, of the volumetric profile, or of the security profile. Mitigation or resolution of faults requires the availability of support for a reconfiguration back to an efficiently working system.

GEMOM is able to rectify such vulnerabilities to faults by dynamically deploying a new instance of the messaging system. GEMOM is resilient and able to utilize redundant modules, hot-swap or switch-over without information loss. These resilience-features allow specialist, independent system actors (viz. watchdogs, security and situation monitors, routers, and MOM clients) to remove or replace compromised nodes from the broader network instantly and without compromising higher level functionality and security.

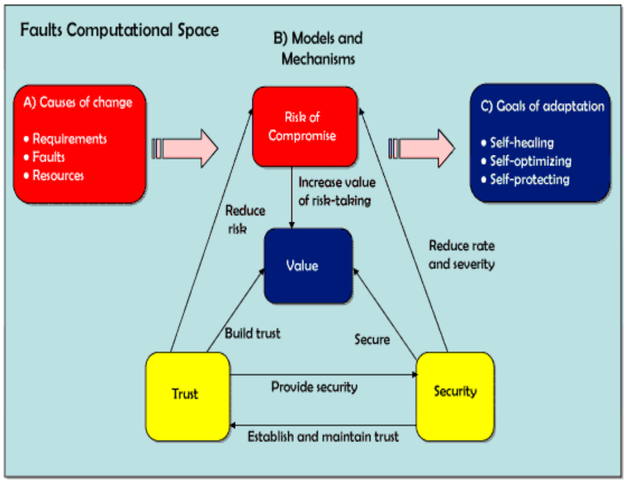

Figure 2. GEMOM Fault Computational Space

The GEMOM fault computational space in terms of risk, trust and security is shown in Figure 2. The space can be categorized as:

A) Causes of change are due to system complexity and environment, in which the GEMOM system has to deal with and needs to adapt to them. These include requirements, resources and faults. Fault tolerance ensures availability by guaranteeing maximum continuity of a service and an acceptable level of service when faults occur. Since the concept naturally lends itself to adaptability, fault and intrusion tolerance mechanisms can be used to increase the availability of a system. At an abstract level within the maintenance of high levels of QoS, reliability and resilience in the presence of threats to critical infrastructures, resources are sometimes restricted in terms of financial budgets and computing infrastructure. A comprehensive risk assessment might therefore be appropriate to assess what components and/or levels require the highest level of protection in order to reduce the overall effect, or at least to mitigate, the worst threats of those events whose effects it would take the longest time to recover from. The ideas within GEMOM can play a considerable role in this area as some of this intelligence can be covered by Service Level Agreements (requirements) through the use of Artificial Intelligent approaches such as naive Bayes and conditional probabilities to predict where within the overall system the highest level of risk - in terms of cause and effect - might be concentrated.

B) Models and mechanisms: A model here means a set of functions used to describe features of either a particular element or multiple elements of a client, e.g., the range of a particular element's possible values, the probability distribution of the number of sub-elements, or the presence and absence of a subset of elements, the structure of elements in a message body. Our models here consist of risk, trust and security adaptations with associated algorithms. Risk is an inherent part of any security or trust system. Risk adaptive security is an emerging technology that adapts its decision based on a computation of security risk. Trust is a necessary prerequisite basis for a decision to interact with an entity. Trusting an entity is always associated with risk since there is always a chance that the entity will behave contrary to expectations. Trust reduces risk, builds confidence in the value of a business and provides security. Security supports the process of establishing and maintaining trust through the provision of a secure and trustworthy environment. Security also reduces the rate and severity of compromises by continuously adjusting and responding to constantly emerging and changing threats. Based on the above described relationships of cause and effect as a foundation, GEMOM adapts and combines adaptive risk-based security, trust-based security, and security-based trust. The effect of this combination is to increase the strength of security and the degree of trust in the messaging system, and to reduce the rate and severity of compromises [59]. Many promising approaches exist, bringing together tools from control theory, biology, economics, utility theory, artificial intelligence, etc. A model here means 'a set of functions used to describe features of either a particular element or multiple elements of a client, e.g., the range of a particular element's possible values, the probability distribution of the number of sub-elements or the presence and absence of a subset of elements, the structure of elements in a message body'.

C) Goals of the adaptation (self-healing, self-optimizing, self-protecting) are adapting topology, resource usage, 'fidelity', etc. The self-healing capabilities can prevent and recover from failure by automatically discovering, diagnosing, circumventing, and recovering from things that might cause service disruptions. The self-optimizing capabilities enable the system to continuously tune itself – proactively to improve on existing processes and reactively in response to environmental conditions. Its self-protecting capabilities enable the system to detect, identify, and defend against viruses, unauthorized access, and denial-of-service attacks [12]. The driving factors and the needs for dynamic adaptation can be summarized from [59] as follows. The driving factors for adaptation are (i) the convergence of advanced electronic technologies (wireless, handheld, sensors, etc) and the Internet, (ii) the promise of instant access to data and computing no matter where or when, (iii) the changing nature and behaviour of the environment, and (iv) the need for systems to operate in the face of failures and attacks. The need for dynamic adaptation is due to (i) the heterogeneity of hardware, network, software, etc., (ii) the dynamics of the environmental conditions, especially at the wireless edge of the Internet, (iii) the limited resources (such as battery lifetime), and (iv) the software adaptation technologies for detecting and responding to environmental changes, and strengthening self-auditing capabilities of 'always-on' systems.

E. *Self-Adaptive Agent System*

The combination and integration of MOM and self-adaptive agent-based systems in GEMOM, resulting in resilient MOM, render a number of advantages. The self-adaptive agent has the following properties [61]: (i) autonomy, which allows it to operate without the direct intervention of humans or other external systems and to have some kind of control over its actions and internal state, (ii) social ability, which allows it to interact with other agents (possibly humans), (iii) reactivity, which allows it to perceive its environment and respond in a timely fashion to changes that occur in it (the environment), and (iv) pro-activeness, learning, and adaptiveness, which allow it to exhibit goal-directed behaviour by taking the initiative, to learn when reacting and/or interacting with its external environment, and to modify its behaviour based on its experience.

Consequently, the GEMOM components incorporating all these properties have self-adaptive behaviours, built-in capabilities for autonomous operation, monitoring their environments, reasoning, and communicating with other agents and human users. Self-adaptive systems require high-level dependability, robustness, adaptability, and availability. GEMOM meets these requirements by reaping the benefits of agent-based message brokers and overlay nodes: reliability via self-healing, performance via self-adaptation, security via self-protection.

IV. Self-optimization, Self-healing and Scalability

In this section, we investigate the self-optimization, self-healing and scalability functionalities of GEMOM.

A. *Optimizing Security and Protecting Networked Systems*

In GEMOM, self-optimization means making run-time adjustments to the operation of the system so that the values of certain selected operational parameters meet, or get closer to, their preferred range. Typical parameters are the usage of bandwidth, computational power and speed of message delivery.

GEMOM allows for the persistence of an optimized setup: new sessions can be established over a newly evolved topology. Redundancy can be used as a safety measure to secure continued, uninterrupted operation in cases of hardware failure or overload, or a DoS attack, yet being utilized at the expense of computing power, hardware and bandwidth. Self-healing can be seen as a ‘sibling’ of self-optimization, where the structure of running nodes, tasks, and communication paths are adjusted as a response to failure-type events, in order to re-establish an initial system structure equivalent pattern. Equivalence in this case assumes functional, resilience and security parameters.

Optimization can be achieved by autonomous agents inside each node, by a central agent for the entire system, or a hybrid approach. The knowledge of the autonomous agents is typically limited to the node itself and its immediate neighbours. These agents normally follow a set of empirical rules that are known statistically to make the network perform reasonably well if all nodes adhere to them. If a central agent is deployed it will have knowledge about the entire network and all the communication paths, and thus be better equipped to make decisions that are globally optimal. The system is then, however, exposed to attacks or failures that could disrupt the communication between the agent and one of the nodes, whilst systems based on autonomous per-node optimization agents are more robust and self-healing than centrally managed systems.

B. *Evolution Algorithms*

The GEMOM system utilizes one overall managing entity per Broker core, running on the same host platform. The manager performs both optimisation and healing in terms of starting a replacement broker in case of malfunction. The manager runs as a parent process of the broker core. The manager process manages routing and group replication. This is based on communication with other brokers. Evaluation algorithms have been developed to decide optimal values for various metadata and routing properties, balancing considerations for:

Manageability: Each node performs a limited and well-defined set of functions, and only has responsibility for a manageable number of groups of nodes.

Scalability and resilience: In a system of cooperating brokers, publishers and subscribers, there have to be sufficient replication of paths and messages to avoid overloading specific servers, and to be able to sustain random and sudden fallout without interruption of service.

Economy: A system of co-operating nodes has to use as little bandwidth and hardware resources as possible.

Consequently, two approaches are used to achieve resilience and evolution in GEMOM, one being the management of reserve resources in such an overlay network, the other being empirical correlations.

C. *Redundant Publishers*

Published messages often originate from outside of the environment when a user is subscribing to the messaging system. They are received through feeds that can be compromised. GEMOM allows for the application of redundant feeds sourcing data from the same or different provider (publisher). Switch-over is instantaneous with minimal loss of other features providing feeds are compatible in terms of capabilities.

D. *Quality of Service*

GEMOM as middleware is well-suited to provide an abstraction for QoS towards the application. GEMOM addresses the QoS requirements of applications through service level agreements, which are managed by the middleware. The supported QoS metrics and parameters include message latency, transaction rate, loss rate, delivery semantics, message ordering, message delay variation, and expiration time. The management of the QoS requirements given by an application is performed by the same regime that manages security-related properties, i.e., using the extended concept of faults.

E. *Scalability and Resilience*

In GEMOM, scalability and resilience are achieved via co-operating message brokers, publishers and subscribers with sufficient replication of paths and namespaces, and clustering of topics into groups of one or more, with group replication. This allows the system to avoid overloading specific brokers, and to sustain random and sudden fallout without any interruption of service.

For scalability with respect to message volume, GEMOM provides switch-over to redundant components preserving, and not compromising, scalability.

V. Holistic and Systematic Adaptive Security

The GEMOM [7] system is a resilient and scalable MOM that supports adaptive security-management by a monitoring functionality based on security and QoS metrics. Adaptive security in GEMOM refers to a security solution that learns, modifies existing functions, and adapts to the changing threat environment without sacrificing too much of the efficiency, flexibility, reliability and security of the system.

A. Tangible and Demonstrable Improvements in Security

Within the current MOM technologies the security requirements are somewhat rigid and do not form an integral part of the overall capability in a scalable and flexible way. Unfortunately, the state of the art in developing credible and sufficient security requirements in a holistic way is still in its childhood. Improvements to security are through a security monitoring system supported by appropriate security metrics, explicit enhancements to the authorization process, the explicit provision of resilience, and the provision of an associated software suite to support the discovery of vulnerabilities in systems that deploy GEMOM. Multiple modes of authentication and management of the authentication strength during authorization processing and fine-grained authorization of GEMOM usage at broker, cluster, topic or message level are provided.

B. Adaptive Authentication and Identification system

Authentication in GEMOM is based on multiple, possibly redundant, mechanisms and may include passwords, smart cards, uni-modal biometrics, and their fusion. This solution allows for interoperable context-sensitive security mechanisms, where the security mechanisms adjust to the needs defined by applications and the security level requested by the transaction. The authentication mechanism developed in GEMOM makes use of authentication policies that can be dynamically adapted according to needs, for example to take into account application needs for authentication security level. Appropriate security metrics are being developed to offer evidence for the adaptive security management. Flexibility is achieved by adding a normalized strength of authentication to the actor, before it is authorized as a pair (*Actor*, *Authentication Strength*), the *Authentication Strength*, which is an aggregated metric that depicts the overall security level of the authentication solution.

GEMOM Identity and Authentication Management Components: as depicted in Figure 3, the GEMOM identity & authentication management has the following functional components:

- A GEMOM Identity Provider (IdP) Service in charge of managing all identity related data, users' authentication and the provision of an entities' Attribute Service, that is of a set of functionalities through which additional entities information can be searched and provided. The GEMOM IdP Service is in charge of managing trust management relationships between GEMOM systems and other systems with which a federation is configured. The GEMOM IdP Service also provides the capability of using different back-end silos so that existing entities data sets can be reused;
- A GEMOM Authentication Service Client to be used to access the GEMOM IdP Service for all entities authentication and attribute needs. As depicted in the figure the GEMOM Authentication Client has to be used both by the GEMOM Message Broker functional component, as well as from GEMOM application clients. The GEMOM Authentication Service Client takes care of managing all interactions with the GEMOM IdP Service, select the right authentication protocol, translation of security tokens, as well as submission of attribute queries and acquisition of entities attribute values.

As indicated in Figure 3, end-users can use different identity technologies and tokens, leaving to the GEMOM Authentication Service Client the job of properly managing the corresponding protocols, data and transactions.

Figure 4 provides a more fine-grained view of the functional components involved in end-user authentication and of the kinds of authentication credentials the end-user has at his/her disposal (username/password pair, X.509 [62] certificate, and smartcard).

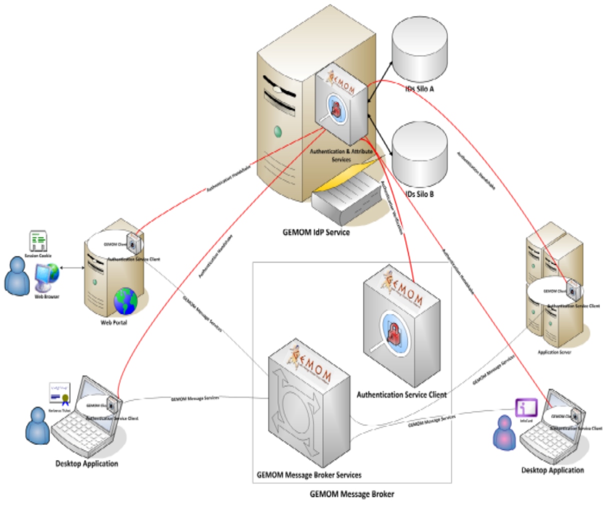

Figure 3. Identity Management and Authentication architecture

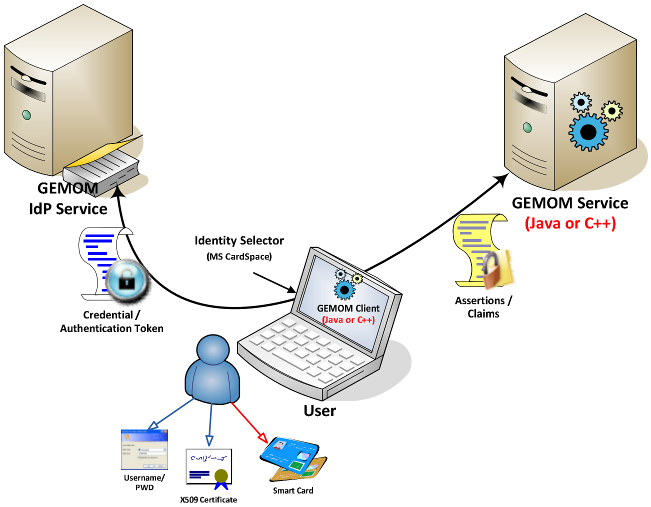

Figure 4. End-user authentication components

C. *Adaptive Authorization*

Authorization in GEMOM supports access rights to namespaces, cluster groups, topics and single messages, the application of access rights to a single message being the smallest level of granularity to which authorization rules can be applied. The GEMOM authorization model also supports multiple user roles, defining access rights and varying performance and reliability requirements depending on the type of user. It is the pair (*Actor*, *Authentication Strength*) by which the actor was authenticated that is a unit-entity that GEMOM authorizes. The GEMOM authorization process is carried out using this pair as a basic composite key, taking into account the following: (i) each user belongs to a group, and the basic strength of the user-authentication key is translated into a vector of strength of group-authentication pairs, (ii) the system is perceived as having certain multi-dimensional security profiles, and boundaries are defined in each dimension, (iii) an application is divided into an arbitrary set of modules, and an abstract notion of operation on a module is defined in which a module can allow an arbitrary number of operations to be performed on it. Access rights are defined for the pair (module, operation), and (iv) certain groups of users that are authenticated with strengths that fall into certain ranges are allowed to perform certain operations on application modules within certain periods of time, within defined context boundaries and within certain dynamic security boundaries. The development of adaptive features of the authentication, identity management and authorization processes is described in detail in [7][59].

D. *Adaptive Security Monitoring*

The GEMOM Security Monitoring System (SMS) is based on security level estimation mechanisms to enable resilience of the system. These mechanisms utilize security metrics, developed in a systematic security requirement decomposition process, introduced in [60], and enhanced in [48]. The security metrics compare the actual security level to the reference level set by security requirements of security-enforcing mechanisms or security functions [63]. Consequently, the definition of appropriate requirements, which address security, resilience, self-healing and evolution, has been the core activity in the development of GEMOM Adaptive Security Management (ASM) functionality.

The SMS includes measurement data collection mechanisms and interfaces to the system components under measurement, associated adaptive security knowledge repositories, metrics and trust, confidence and reputation information and suitable algorithms for using metrics. The SMS carries out security monitoring and supports ASM operations based on the on-line security metrics. The SMS is connected to the GEMOM Message Broker, Authentication and Authorization Module, Audit and Logging Module, QoS Accessory Module, Anomaly Detector Module, and memory elements, storage and network interfaces. In addition to the logs produced by the Message Broker, the monitoring system is able to monitor messages and metadata. Figure 5 depicts an example GEMOM subnet and information flow relevant to the SMS.

The collection of Basic Measurable Components (BMCs) of the security metrics for the GEMOM Security Monitoring System have been introduced in [48] along with a security metrics development methodology, analysis of its benefits and shortcomings and a framework for calculating trust, confidence and trustworthiness of the metrics. BMCs are the leaf components resulting from the security-requirement decomposition, an abstraction for a more detailed development of security metrics.

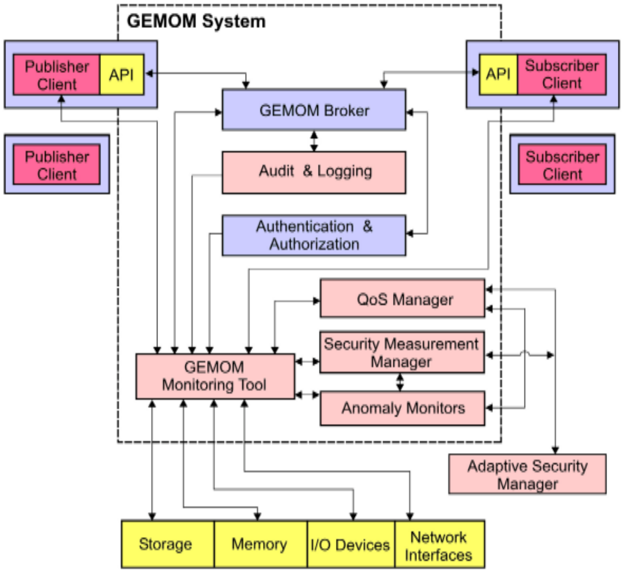

Figure 5. An example GEMOM subnet [64]

Metrics, rules and reputation information are configured by using the Monitoring Tools (MTs) user interface and are stored in a special database. An MT is connected directly to GEMOM Broker(s). Connection between the Monitors and Brokers is arranged via GEMOM Client Interface (GCI), see Figure 6 [65]. Other modules use the GEMOM PS mechanism for communication: publishing and subscribing to relevant topics in a *measurement namespace* [64]. Using this mechanism, the MTs connect to Authentication and Authorization modules, QoS Managers, Anomaly Detector modules, Security Measurement Managers, as well as relevant-use and free-memory entities, storages (hard disks, memory sticks), network interfaces and Input/Output devices (e.g., keyboard).

The following attributes form the minimum set of needed configuration parameters: metric ID, input and output data of the metric, metric calculation formula or heuristics, threshold value(s), and timing information. At the managerial G-Nodes level, the Monitoring Tools co-operate with the ASM. The ASM monitors security, analyses its details, plans adjustments, and executes the planned adjustments through a global control loop, using both manual and automated information. Thus, the ASM manages the behaviour of the overall system from the security point of view. The monitor modules can be updated and enhanced, and new modules can be integrated during runtime operation, supporting the ASM.

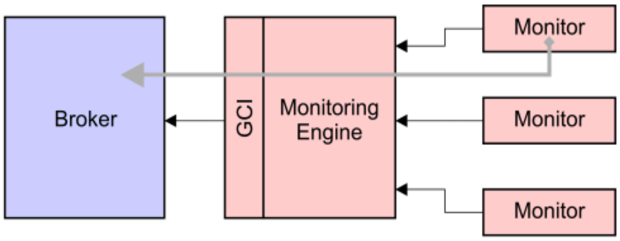

Figure 6. Communication between the Broker and the MT [66]

The GEMOM SMS and the ASM utilize a holistic State of Security (SoS) concept [64][65]. SoS is a time-dependent estimate of the system's security performance level based on the appropriate collection of security metrics that is calculated initially and when triggered. The concept can be used to configure the management of the used security metrics. There are five steps in the estimation process of the SoS:

1. Definition of the initial SoS is done using appropriate security metrics.
2. The current SoS is measured whenever triggered by a timer, an attack, an anomaly or a manual request.
3. Past and current SoS is compared to offer input to the trend estimation in decision-making.
4. The initial SoS is adapted according to decisions made by the ASM functionality.
5. A future SoS is predicted to enable proactive Adaptive Security Management.

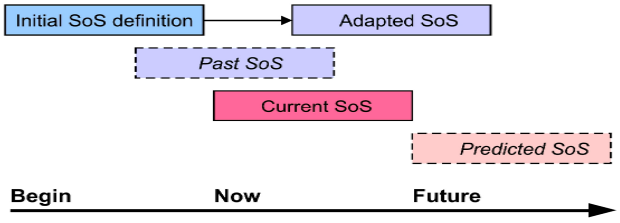

Figure 7. A timeline visualization for SoS estimates [64]

Figure 7 depicts the visualization of the different types of SoS estimates. The predicted SoS is based on the analysis of the past history of the SoS levels and threat and vulnerability trends. The predicted SoS is useful when carrying out proactive operations to ensure the high resilience of the system.

E. *Self-Protection*

A self-protecting system, as defined by IBM [66], can anticipate, detect, identify, and protect itself against threats, unauthorized access, and denial of service attacks. GEMOM as an autonomic MOM has to implement self-protecting capabilities that can detect hostile behaviours as they occur and take corrective actions to make the system less vulnerable. In the GEMOM setting (see Section III), the self-protection is managed either at a single entry point (a micro property), which gives each node authorization, by a coordinated defensive group attack of the other nodes alone (a macro property), or by a combination of the two (defence-in-depth). Figure 8 shows these entry points and their properties.

Most intrusions can be managed by triggering a one-shot behaviour of the GEMOM system. However, the GEMOM system has constantly to be alert, so the degree of protection over time (ongoing) is important [59]. The proactive identification of and protection from, arbitrary attacks are achieved via the combination of anomaly-based self-protection [40], and security monitoring and measurement.

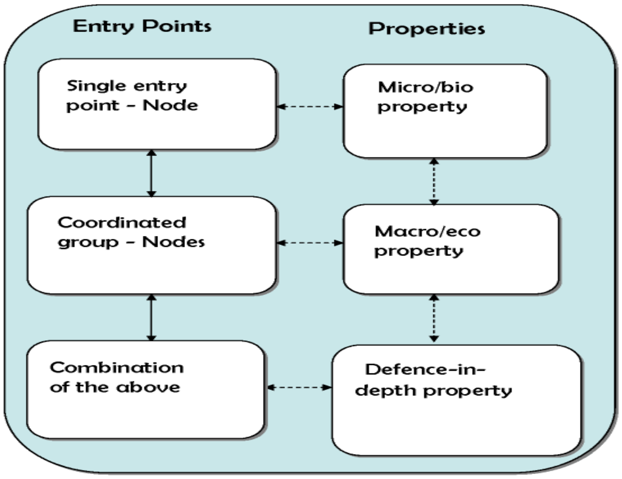

Figure 8. Self-protection entry points and properties

A key component in self-protection is the integration of mechanisms to support the detection of anomalies such as high message rates, degradation of broker performance, e.g., in the context of DoS, and of services to support the detection of anomalous message content in appropriate cases. Detectors are divided into different functions e.g., link-state detection, message-rate computation, bottleneck detection, and overall system representation. For example, the detection of and reaction to, link-state faults, brokers continually send probes to its clients and peers. Metrics such as loss rate and message delay are measured from probes. However, when anomalous metrics are detected, the frequency of probing is increased. A relevant action for example, when a sequence of probes are lost or metrics are above the acceptable range, is that another broker is set as a relay broker between the nodes experiencing a faulty connection in an attempt (not assured as the topology of the underlay is probably not known to the MOM system) to circumvent the failure. In a worst case, if a relay broker can not be found, then the workload of this original broker needs to be taken over by a pre-allocated mirror broker. The distinction between link-failure and broker-failure is established by monitoring between brokers either over the subnet or over what are expected to be disjoint paths in a wider network (Whether a subnet or a wider network is used to support the overlay depends on the application and the nature of the resilience required.). This probing approach maintains a view of the whole system, and scales to a limit of tens of nodes, which is considered enough for our applications. Different approaches to bottleneck-detection are under investigation. The component being integrated currently uses Markov models to predict the values of different individual measurable resources of the broker (broker CPU, message rate, subscription rate etc.) and uses a Naïve Bayes classifier, trained on system operational data, to detect a bottleneck based on the predictions.

The optimal allocation of the workload among brokers (see self-healing later), and redundant mirroring provides an enhanced toleration of burstiness from Flash Events (FE) and DoS attacks. DoS detectors are distributed among the overlay nodes to localize and mitigate DoS attacks. The analysis is achieved by collaboration between overlay nodes. The detection is employed both locally in each node, and by globally monitoring the correlated measurements. We adapt new detect and defence mechanisms to the architecture and context of our federated PSMOM, and some of the mechanisms being based on previous attempts for DoS defence, e.g., [67][68].

Anomalies in the messages in a MOM system can be caused by attacks that propagate through or target on the system. In the PSMOM domain we can profile the normal messages based on the collective characteristics among messages from similar topics, and detect outliers. We are employing a multi-model approach [69], and the models profiling normal messages are chosen based on the system requirements, e.g., whether content is encrypted or not for inspection, and statistical characteristics of messages.

Different detectors with different levels of functions can raise alarms simultaneously. The Management Layer will correlate these alerts and choose a proper reaction. An Overlay Manager is responsible for a range of functions to improve performance and resilience at the management layer. For example, a link fault with long delay and high loss rate might be accompanied by a simultaneous DoS attack alert in this case the Overlay Manager will prioritize the response to DoS attack and suppress the link fault alert for a short time. Experiments based on DoS attacks are being created and the detection and reaction mechanisms validated.

Figure 9 depicts the GEMOM data collection architecture. The sensors and detectors are distributed both inside the GBroker, monitoring messages and extracting features of each topic.

Inside the GBroker, data collection is performed for each topic by computing and updating it during the message-processing stage. An anomaly in a topic can be poisoning the entire cluster. Topics in the same cluster (i.e., topics in the same messaging path connected by the switching GBrokers) will also exchange anomaly detection information through the GBrokers. This can be seen as a simple form of dynamic-taint analysis. That is why there should be a cluster-correlator to decide on the actions of the whole cluster (i.e., Cluster Correlator). This suppresses any actions proposed by the individual switching GBrokers in the cluster, and replaces the actions with the actions decided on by the cluster manager. Correlation of anomalies between switching GBrokers in a cluster is performed by a single cluster correlator, which can be centrally located, or elsewhere, e.g., the last switching GBroker in the path of the cluster. The consumer and producer clients can only send their information to the cluster correlator, which sits on top of the base anomaly detector in the GBroker. The cluster anomaly detector is the anomaly detector that is responsible for taking decisions, and it has the capability to override/suppress reactive settings in the decision-making policies for the topic of the individual GBrokers. Reactive actions may be necessary at individual GBrokers, as some anomalies may be so clear cut and so dangerous, that to wait for the decision of the cluster correlator may be too late. The location of the Correlator is outside of the GBrokers, as shown in Figure 9.

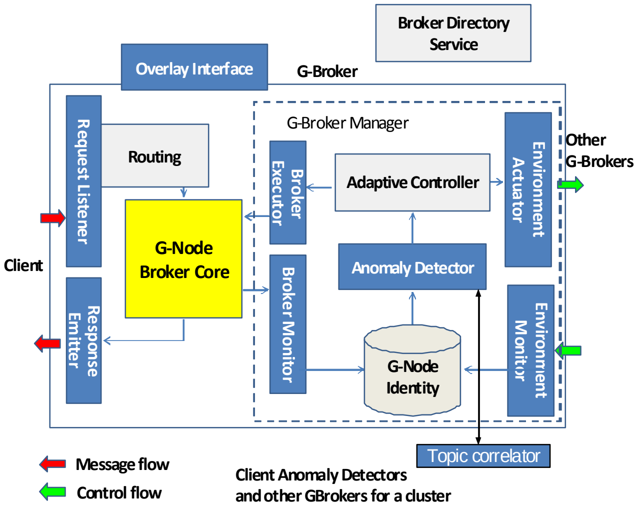

Figure 9. Data collection in context of the GEMOM Architecture [1]

F. *Pre-Emptive and Automated Run-time Vulnerability Management*

Including errors to force a reaction from the system can be a way of making the system more robust. Mechanisms to induce error conditions in the services and to expose vulnerabilities before they happen have been developed. The GEMOM vulnerability management toolkit has been developed to detect vulnerabilities in the deployed GEMOM system as configured by the users. To identify previously unknown faults and loopholes, effective techniques to generating inputs that induce failures have been developed. This is called fault injection or fuzzing technique. Compared with traditional software-testing techniques, fuzzing has been found effective and cost-efficient. It is becoming a legitimate aspect of robustness and security testing.

There are several available fuzzing software and libraries for widely used protocols. However, considerable research is needed into the construction of fuzzers that can capture, without too much manual intervention, the diversity of applications associated with it. Each situation, protocol, or application causes new issues that need to be addressed. The best approach to testing varies between projects. The experience of the testers has a significant bearing on the efficacy of the testing [70].

G. GEMOM Software Security Assurance in General

A secure and resilient system solution based on best practices is not a sufficient security solution by itself. The security, trust, dependability and privacy requirements of a system and applications must be comprehensively analyzed from a security risk perspective and adequate security assurance methods must be used. Security assurance includes a wide variety of activities from security analysis to security testing and monitoring [64]. In recent years, the understanding and tools for security assurance have developed in leaps and bounds, enabling the functional testing and monitoring of security to be part of the normal product development and maintenance processes. Comprehensive risk-aware security analysis guides testing and monitoring activity. Security analysis may include: the investigation of threats, the specification of security requirements, the modelling of attack, the investigation of vulnerability, and the assessment of security level and performance using adequate security metrics. Most of the on-line and off-line security metrics developed for GEMOM can be utilized for the security assurance.

H. Self-Healing

The atomic unit to offer resilience is at the topic level. For performance reasons, QoS monitoring in GEMOM is at the namespace level by default, though monitoring at the topic level is possible; this is particularly relevant for monitoring anomalous individual message content when appropriate. Namespaces are also further grouped into items, to support scalability of the resilience decision-making [71].

The mirror and relay concepts are used in resilience management. The primary namespaces are the namespaces that are handled by the GBroker. A GBroker will also act as a mirror for other broker namespaces and these namespaces form the mirror namespace set, also held at the broker. A primary namespace tree can be partitioned into many sub-namespaces for mirroring, and each sub-namespace assigned to a different GBroker for mirroring. The namespaces assigned to a broker for mirroring is called the mirror namespace set.

In practice, assured delivery is a common requirement of MOM users. This means that if a subscriber loses connection (e.g., through border gateway failure between the broker and the subscriber, or subscriber site failure) then the MOM has to retain messages until the subscriber can later pick up the messages. A time limit can be put on the retention in the MOM, but the relevant policy is application dependent. This means that if a broker has to take over the function of all or part of another broker then a lot of state information may need to be available. This also means that, in contrast to working on P2P systems, it is sensible to pick a mirror candidate prior to failure and not delay the decision of alternatives till the time of failure so that state can be tracked.

There are three steps. First, the whole workload of namespaces is partitioned into items, and each item is a subset of namespaces. Items are disjoint. Assuming those items are known, we are interested in the allocation or re-allocation of such items to each broker in the MOM system. Second, an optimal allocation of items over the overlay is determined. A combinatorial auction mechanism has been implemented for finding the optimal combinations of items to be allocated to each broker (i.e., the winner determination problem). The brokers act as bidders and bid for sets of nodes and the MOM system acts as the auctioneer. In a complex problem like providing resilient service, this auction based allocation mechanism gives brokers some degree of freedom in applying different preferences to choose the items they bid on. The system is able to find an optimal solution, from possible combinations of all the bids placed by brokers. By optimal combinations we mean the best allocation of items, where the risk of brokers being saturated is estimated to be the least, and where the chance of brokers generating the maximum revenue to the system is estimated to be optimal. The price to bid is based on the risk function that estimates the probability of exceeding the GBroker's capacity by exploiting the correlation between different items using the variance covariance matrix of the namespaces of the workload. Since positively correlated items have a super-additive effect on consuming resources of the system, the bidding function put preference bidding on non-positively correlated item combinations that posses less risk.

The third step is to provide redundancy and reactive solutions to adapt to system faults and degradations. After the initial allocation of the workload in order to react to possible failures and service degradation, we compute solutions to re-allocate workload and introduce redundant mirror items with available resources. This is done by applying either extra rounds of auctions or by an optimal [67] search again based on the risk function. The solutions are saved in a case data base to support timely reaction.

VI. Enhanced Interoperability and Integration

A PS MOM-based system can be modelled and re-factored with ease at run-time as well as at design time. The exchange of messages is connectionless and asynchronous. The PS MOM system is inherently extensible, etc. These features make PS MOM a powerful base for resource efficient implementation of scalability, resilience and management of vulnerabilities in a distributed system. GEMOM as a PS-based MOM has these properties.

GEMOM further supports better interoperability and integration of information systems by allowing actual instances to be configured, so various functions are subcontracted to one or more separated external or federated entities. This separation allows the use of different security layouts for different individual services or clusters of services. The most important advantages of this approach are:

- Focus of different functional clusters on different issues and core competences. For example, consider the dynamism of messaging. The rapid changes in message volume is often such that, in terms of economy of the offered solution, it is common to separate pure messaging, authentication and security related services. Pure messaging (e.g., without bundling) in highly scalable environments can be very resource intensive;
- Message brokering is not compromised while an incident is flagged on one or more security monitors awaiting resolution;
- Security functions (e.g., authentication, authorization, key management, security metrics processing) can be implemented to far higher standards and be less resource intensive by separating them from the other parts of the system;
- Non-intrusiveness of the monitoring system: the monitoring system does not cause any harm to the normal operation of the measurement target system and does not affect the measurement results; and
- Bridges and adapters for industry standard messaging systems: GEMOM deploys a framework for integration with other messaging platforms and information service busses. GEMOM is interoperates with platforms such as JMS [72], Tibco's RV [73], Reuter's Triarch, and IBM's MQ Series [74] through the provision of bridges and adapters.

VII. Adaptive Integration Architecture

The GEMOM framework includes adaptive integration functions and tool-sets. This section briefly describes the integration of these tool-sets using the Adaptive Security Manager (ASM) as an example of how these tools can be integrated. Figure 10 shows the adaptive integration architecture.

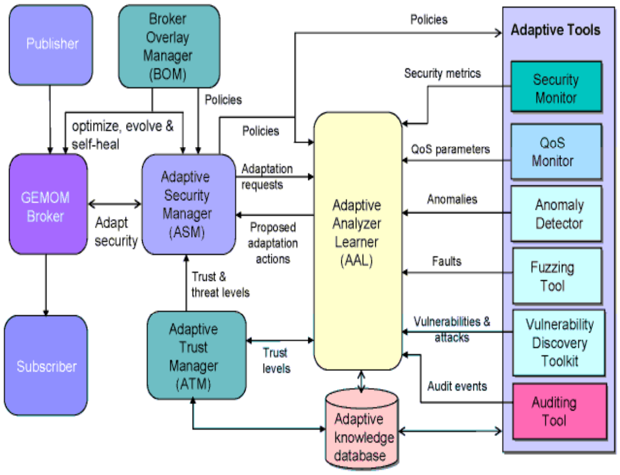

Figure 10. Adaptive integration architecture

The Broker Overlay Manager (BOM) provides resilience and adaptability by utilising an overlay network of publish/subscribe MOMs and can function despite lack of privileged knowledge of the underlying infrastructure. Figure 11 shows the conceptual structure of the BOM. As shown in Figure 11, the BOM uses models for supporting resilience, optimisation, evolution, QoS and security. It provides two methods for interactions with the models: (i) model can be a plug-and-play rule loaded into BOM process, and/or (ii) model can be hosted in an independent application and communicate with BOM over dedicated GEMOM message broker. It also provides global mechanisms (such as global policies for adaptation, optimization, evolution, and self-healing) for models' functionality to be able to alter the behaviour of the GEMOM system at the level of machine, broker, client, namespace or topic.

The BOM performs autonomous adjustments to the run-time configuration of the system in order to preserve and maintain optimal and uninterrupted operation, recover from partial breakdowns and use newly acquired information about the topology of the system and its surrounding in order to evolve into better system. It considers the provision of self-optimisations of the system both at the level of the overlay manager and at the sub-components of the overlay manager. In the context of MOM system, different approaches are used to support resilience, multi-path redundancy, reactive routing, path disjointedness and namespace mirroring. A Case Based Reasoning (CBR) approach that allows reactive response was also developed, where the case base is created by repeated use of the optimisation process. Additionally, an approach based on the analysis of system correlations is proposed as a technique to respond to a complex dynamic system.

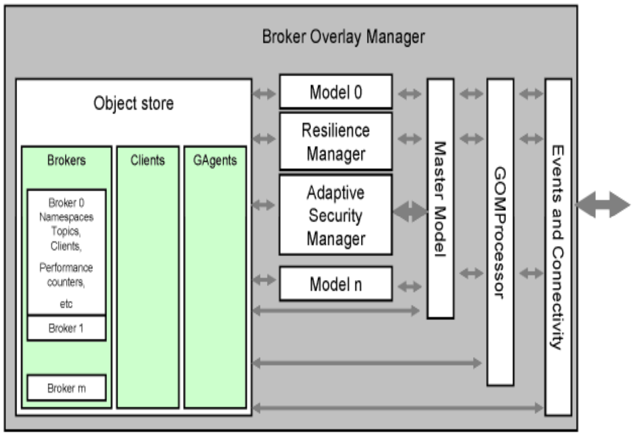

Figure 11. Conceptual structure of the Broker Overlay Manager

Finally, the BOM supports mechanisms for adding G-Nodes; for measuring QoS between overlay components, publishers and subscribers and deciding what action to be taken to mitigate loss of QoS or breakdowns; for discovering and communicating with other components in the overlay network; for evaluating the performance of the system in the context of the monitored performance; for establishing the state of the overlay network; and for making decisions on the reconfiguration of routing and message passing. It also learns from experience and uses its new knowledge in its prediction and decision-making.

The ASM [59] manages and controls all the security components as an integrated GEMOM security infrastructure. Its security services adapts to the rapidly changing contexts of the GEMOM environment. The ASM model consists of a continuous cycle of monitoring, assessment, and evolution to meet the challenges in the changing environments and threat situation. It utilizes contextual information and decision-making to select the 'best' security model for a given situation. The ASM includes the integration of adaptive control loop functions (monitoring, analysis and response), and tool-set, elastic-fine-grained adaptive authorization, adaptive authentication and federated identity management, and tools and processes for pre-emptive vulnerability testing and updating. While each component implements a local adaptation control loop, the ASM implements a global adaptation control loop. Here the sensors are Anomaly Detectors, Security Monitors, Fault Detectors, QoS Monitors, and Auditing and Logging.

The ASM component provides adaptive security and trust through changing security policies, algorithms, protocols and encryption schemes according to context parameters, such as environment, system threats, user threats, trust levels, usage, security and trust metrics, faults, and quality of service. Fault and intrusion tolerance mechanisms are used to increase the availability of a system, and previous faults caused by the user are used to increase suspicion-level. The system threat-level and the user suspicion-level are maintained by and obtained from the Adaptive Tools (like Security Monitor, Anomaly Detector, and Fuzzing Tool). Figure 10 depicts the relationships between the ASM and other components.

The Adaptive Analyzer and Learner (AAL) component implements the analysis function of the adaptive control loop and analyses the collected information using established analysis and decision-making methods. It processes the collected data, along with other information (e.g., security policy, threat levels, or trust levels boundaries) and proposes actions to bring about a new stage. The Adaptive Tools sense and gather contextual information both from within the system and from the environment. They distribute information about the security environment to the AAL and adaptive database. The Vulnerability Discovery Toolkit allows the identification and understanding of the risks and vulnerabilities of the GEMOM system and the forming of trust solutions to address the risks and vulnerabilities. The Fuzzing Tool allows an effective black box testing technique to be used for finding security flaws from software.

The Adaptive Trust Management (ATM) model [59] is a compromise-based trust model that provides information about any attack on the system and the nature of that attack for the purpose of establishing whether, and if so, how different properties of the system have been compromised. In addition, it establishes whether these properties can be trusted for a particular purpose in spite of being compromised and to what degree these judgments should be suspected or monitored. It also incorporates a framework for calculating trust, confidence and trustworthiness of the trust and risk impact metrics.

This adaptive integration demonstrates GEMOM's solution that consists of a continuous cycle of monitoring, measurement, assessment, optimization, self-healing, adaptation and evolution to meet the challenges in the changing environments by (i) provision of self-optimisations of the system both at the level of the overlay manager and at its sub-components, (ii) combining adaptive risk-based security, trust-based security, and security-based trust, and (iii) integrating different metrics, assessment and observation tools.

VIII. GEMOM Prototyping and Validation Results

The GEMOM project has prototyped: a full featured message broker, transparent completion and encapsulation publishing framework, adaptive security implementation (such as authentication, authorization, key management, and identity management), MOM Intelligent Fuzzing Tool, Security Monitoring Tool, and configuration and deployment of management and development process tools. The project has also developed the following demonstrators: Interfaces for enhanced resilience, QoS and security, security and QoS monitoring system, Integrators with well-known commercial MOM systems (JMS, Tibco's, Reuters, and IBM's MQ Series), and Broker Manager Agent without and with optimization.

These GEMOM prototypes have been validated in five case studies: a collaborative business portal, a dynamic linked exchange, a financial market data delivery system, a dynamic road management system, and a banking scenario for transaction processing. This validation and evaluation process allowed the core GEMOM platform and innovations to be tested. Different scenarios represent differing specific requirements and needs. GEMOM maintains some core requirements: security, performance, speed, and scalability; as a result, the overall approach through the delivery of all the specified enhancements will achieve these top-level needs. The validation and evaluation within real application use cases means that features and enhancements in terms of guaranteed delivery, security, QoS, and resilience were tested against the specific requirements of each use case.

By looking at a number of diverse applications, GEMOM can be tried for scalability, applicability and effectiveness across a wide set of market sectors. A typical way of thinking about this is guaranteed delivery – in some cases, as long as the transaction is completed in the Banking scenario, the requirement is met. However, in the financial market data scenario, guaranteed delivery can have strong time constraints, i.e., if a message is not sent within a given time period – it becomes redundant and the next one carrying the updated data should be sent. By looking at more than one application scenario, it allows us to test scalability, performance, etc., across many user scenarios.

IX. Conclusions and Future work

In this paper, we have described a self-healing and secure adaptive messaging middleware for business-critical systems that is designed to adapt to dynamically changing environments. This middleware system, GEMOM, makes advances in the areas of resilience, self-healing, self-adaptation, scalability, integrated vulnerability management, better interoperability and integration of distributed business-critical systems, and holistic and systematic adaptive security monitoring and measurement. The combination and integration of MOM and agent-based systems, resulting in resilient MOM, advances the state-of-the-art. The system is capable of autonomously adjusting the run-time configuration of the system in order to preserve and maintain optimal, uninterrupted operation, recover from partial breakdowns and use newly-acquired information about the topology of the system and its surroundings in order to evolve into a better system, improving and increasing the strength of security and degree of trust in the system by combining adaptive risk-based security, trust-based security, and security-based trust, and improving the assessability and verifiability of the trustworthiness of the system by integrating different metrics, assessment and observation tools.

In our future work we plan to enhance the intelligent algorithms to improve the robustness, self-healing, self-adaptive, holistic and systematic assurance of adaptive security of the overall integrated system. In order for governments to fulfil their functions and do their job properly, it is important that critical infrastructures be resilient and secure in order to operate reliably and dependably in the presence of threats to them. The resilience and security of infrastructures are a high-priority requirement for governments. In our future work we intend to address this matter and apply our solutions to meet this requirement.

Acknowledgment

This research is in the context of the EU project GEMOM (Genetic Message-Oriented Secure Middleware, www.gemom.eu), 2008-2010, Grant Agreement No. 215327, approved by the EU Commission. The authors acknowledge the efforts of all partners who participated in the development of the GEMOM framework. Mr. Trenton Schulz of Norwegian Computing Centre, Mr. Petri Heinonen of VTT Technical Research Centre, and Dr. Peng Jiang and Mr. Jinfu Wang of Queen Mary University of London are acknowledged for their valuable and professional comments, and Mr. Cristoforo Seccia of TXT e-solutions Spa for his support and perseverance.

References

[1] H. Abie, R. Savola, and I. Dattani, Robust, Secure, Self-Adaptive and Resilient Messaging Middleware for Business-critical Systems. In: The First International Conference on Adaptive and Self-adaptive Systems and Applications, ADAPTIVE 2009, November 15-20, 2009 - Athens/Glyfada, Greece.

[2] H. Li and G. Jiang, Semantic Message-Oriented Middleware for Publish/Subscribe Networks. Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III. In proceedings of the SPIE. In Proceedings of SPIE,5403:24–133, 2004.

[3] D. Lewis, J. Keeney, D. O'Sullivan, and S. Guo, Towards a Managed Extensible Control Plane for Knowledge-based Networking. Lecture Notes in Computer Science, Large Scale Management of Distributed Systems, Springer Berlin / Heidelberg, 4269/2006 (0302-9743):98–111, 15 October, 2006.

[4] S. Parkin, D. Ingham, and G. Morgan, A Message-oriented Middleware Solution Enabling Non-repudiation Evidence Generation for Reliable Web Services. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 4526/2007(0302-9743):9–19, 06 June, 2007.

[5] ESB, Enterprise Service Bus (ESB). Accessed May 30th, 2010, from http://www.sonicsoftware.com/psm/enterprise-service-bus/index.ssp.

[6] GEMOM, Genetic Message-Oriented Secure Middleware Technical Annex, Grant Agreement No: 215327, approved by the EU Commission, October, 2007.

[7] H. Abie, I. Dattani, M. Novkovic, J. Bigham, S. Topham, and R. Savola, GEMOM - Significant and Measurable Progress Beyond the State of the Art. in Proc. ICSNC 2008, 26-31 October, 2008.

[8] E. Curry, D. Chambers, and G. Lyons, Extending Message-Oriented Middleware Using Interception, Proc. 3rd Int'l Workshop on Distributed Event-Based Systems (DEBS 04), 2004, pp. 32–37.

[9] D. Ghosh, R. Sharman, H. R. Rao, and S. Upadhyaya, Self-healing Systems - Survey and Synthesis, Decision Support Systems, Volume 42, Issue 4, January 2007, Pages: 2164-2185.

[10] M. Morandini, L. Penserini, and A. Peri, Towards Goal-oriented Development of Self-adaptive Systems. Proceedings of the 2008 international workshop on Software Engineering for Adaptive and Self-managing Systems, Pages:9-16, Leipzig, Germany, 2008.

[11] R. Laddaga, Self-Adaptive Software Problems and Projects. In SOFTWARE-EVOLVABILITY '06: Proceedings of the Second International IEEE Workshop on Software Evolvability (SE'06), Washington, DC, USA, 2006. IEEE Computer Society, pp. 3–10.

[12] D. M. Chess, C. C. Palmer, and S. R. White, Security in an Autonomic Computing Environment. IBM Systems Journal, Vol. 42, No 1, 2003 107-118.

[13] O. Kramer, Evolutionary Self-Adaptation: A Survey of Operators and Strategy Parameters. Evolutionary Intelligence, Springer Berlin / Heidelberg, Saturday, February 06, 2010.

[14] R. Hinterding, Z. Michalewicz, and A. E. Eiben, Adaptation in Evolutionary Computation: a Survey. In Proceedings of the Fourth IEEE Conference on Evolutionary Computation, Indianapolis, IN (1997), pp. 65-69.

[15] D. Miorandi, L. Yamamoto, and F. D. Pellegrini, A Survey of Evolutionary and Embryogenic Approaches to Autonomic Networking. Computer Networks, Volume 54, Issue 6, 29 April 2010, pp. 944-959.

[16] L. Rodero-Merino, A. Fernandez, L. Lopez, and V. Cholvi, A Topology Self-adaptation Mechanism for Efficient,Resource Location. In Lecture Notes in Computer Science, Proceedings of the 4th International Symposium on Parallel and Distributed Processing and Applications (ISPA 2006). Springer-Verlag, 2006, pp. 660–671.

[17] S. Dustdar, K. M. Goeschka, H. L. Truong, and U. Zdun, Self-Adaptation Techniques for Complex Service-oriented Systems.Fifth International Conference on Next Generation Web Services Practices (NWESP '09), 2009, pp. 37–43.

[18] P. D. Alencar and H. Weigand, Challenges in Predictive Self-Adaptation of Service Bundles. WI-IAT, Vol. 3, pp.457-461, 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, 2009.

[19] E. Gjørven, R. Rouvoy, and F. Eliassen, Cross-layer Self-Adaptation of Service-oriented Architectures. In Proceedings of the 3rd workshop on Middleware for service-oriented computing, 2008, pp. 37–42.

[20] P. Reinecke, K. Wolter, and A. V. Moorsel, Evaluating the Adaptivity of Computing Systems. Performance Evaluation, Special Issue on

Software and Performance, Volume 67, Issue 8, August 2010, pp. 676-693.

[21] G. Giannakopoulos and T. Palpanas, Adaptivity in Entity Subscription Services. 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009, Attents/Glyfada, Greece. IEEE Computer Society, pp. 61–66.

[22] C. T. R. Hager, Context-Aware and Adaptive Security for Wireless networks. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, November, 2004.

[23] A. Shnitko, Practical and Theoretical Issues on Adaptive Security, Novosibirsk State Technical University, and Adaptive security in complex information systems. In Proc. 7th Korea-Russia International Symposium on Science and Technology, (8023863):206– 210, 28 June - 6 July, 2003.

[24] S. H. Son, R. Zimmerman, and J. Hansson, An Adaptable Security Manager for Real-time Transactions. In Proc. 12th Euromicro Conference on Real-Time Systems,, 19-21 June, 2000, pp. 63–70.

[25] P. A. Schneck and K. Schwan, Dynamic Authentication for High-Performance Networked Applications. In Proc. 6th International Workshop on Quality of Service, 18-20 May, 1998, pp. 127–136

[26] J. Zou, K. Lu, and Z. Jin, Architecture and Fuzzy Adaptive Security Algorithm in Intelligent Firewall. In Proc. MILCOM, 2:1145–1149, October 7-10, 2002.

[27] H. Abie, P. Spilling, and B. Foyn, Rights-Carrying and Self-enforcing Information Objects for Information Distribution Systems. Lecture notes in computer science, Information and Communications Security: 6th International Conference, ICICS 2004, Malaga, Spain, LNCS 3269(0302-9743):546–561, 27-29 October, 2004.

[28] H. Abie, Distributed Digital Rights Management: Frameworks, Processes, Procedures and Algorithms. VDM Verlag, 02 October 2009, Paperback, ISBN-10:3639202961, ISBN-13:978-3639202960.

[29] A. Pietzowski, B. Satzger, W. Trumler, and T. Ungerer, A Bio-inspired Approach for Self-protecting an Organic Middleware with Artificial Antibodies. Self-Organizing Systems, LNCS, Springer Berlin / Heidelberg, Vol. 4124/2006, September 21, 2006, pp. 202–215.

[30] I. Djordjevic, S.K. Nair, and T. Dimitrakos, Virtualised Trusted Computing Platform for Adaptive Security Enforcement of Web Services Interactions. IEEE Int. Conference on Web Services (ICWS 2007), 9-13 July 2007, pp. 615–622.

[31] J. Luo, X. Ni, and J. Yong, A trust degree based access control in grid environments. Information Sciences: an International Journal, Volume 179 , Issue 15, July 2009, pp. 2618–2628.

[32] J. Ma, H. Abie, T. Skramstad, and M. Nygaard, Requirements for Evidential Value for the Assessment of the Trustworthiness of Digital Records over Time. In the Proc. of the MASS 2009, IEEE Symp on Trust, Security and Privacy for Pervasive Applications (TSP 2009), Macau SAR, P.R.China, October 12-14, 2009, pp. 796–803.

[33] A. Boukerche and Y. Ren, A Trust-based Security System for Ubiquitous and Pervasive Computing Environments. Computer Communications, Vol. 31, Issue 18, December 2008, pp. 4343–4351.

[34] T. Goovaerts, B. D. Win, and W. Joosen, A Comparison of Two Approaches for Achieving Flexible and Adaptive Security Middleware. In Proc. of the 2008 workshop on Middleware Security, Leuven, Belgium, December 2, 2008. ACM 2008, pp.19–24.

[35] A. Elkhodary and J. Whittle, A Survey of Approaches to Adaptive Application Security. International Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '07), 20-26 May 2007.

[36] S. Sadjadi, A Survey of Adaptive Middleware. Technical Report MSU-CSE-03-35, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, December 2003.

[37] P. McKinley, S. Sadjadi, E. Kasten, and B. Cheng, A Taxonomy of Compositional Adaptation. Technical Report, MSU-CSE-04-17, 2004.

[38] L. Marcus, Semantics of Static, Adaptive, and Incremental Security Policies. First Symposium on Requirements Engineering for Information Security (SREIS) March 2001, Indianapolis, Technical Report ATM 2001(8104-05)-1, The Aerospace Corporation July 2001.

[39] R. W. McGraw, Risk-Adaptable Access Control (RAdAC). September 2009, accessed May 26th, 2010, http://csrc.nist.gov/news_events/privilege-management-workshop/radac-Paper0001.pdf.

[40] G. Qu and S. Hariri, Anomaly-Based Self-Protection against Network Attacks. In Autonomic Computing: Concepts, Infrastructure, and Applications, Ed.: M. Parashar and S. Hariri, CRC Press, 2007, pp. 493–521.

[41] T. Ryutov, L. Zhou, C. Neuman, T. Leithead, and K. Seamons, Adaptive Trust Negotiation and Access Control. In Proc. 10th ACM symposium on Access control models and technologies, 1-3 June, 2005, pp. 139–146

[42] H. Shrobe, J. Doyle, and P. Szolovits, Active Trust Management for Autonomous Adaptive Survivable Systems. Self-adaptive Software, 2000, pp. 40–49

[43] J. Weise, Security Architecture and Adaptive Security. SSA (Information Systems Security Association) Journal, July, 2008, pp. 10–15.

[44] R. M. Venkatesan and S. Bhattacharya, Threat-Adaptive Security Policy. In Proc. IEEE International Performance,Computing, and Communications Conference, 5-7 February, 1997, pp. 525–531

[45] P. Lamanna, Adaptive Security Policies Enforced by Software dynamic Translation. A Thesis in TCC 402 25 March, 2002.

[46] C. Wang and W. A. Wulf, Towards a Framework for Security Measurement. In the Proc. of the 20th National Information Systems Security Conference, Baltimore, MD, Oct. 1997, pp. 522–533.

[47] T. Heyman, R. Scandariato, C. Huygens, and W. Joosen, Using Security Patterns to Combine Security Metrics. In the Proc. of the 3rd Int. Conf. on Availability, Reliability and Security (ARES '08), pp. 1156–1163.

[48] R. Savola and H. Abie, Development of Measurable Security for a Distributed Messaging System. In: International Journal on Advances in Security, Vol. 2, No. 4, 2009, ISSN 1942-2636, pp. 358–380 (Published in March 2010).

[49] M. Schiffman, G. Eschelbeck, D. Ahmad, A. Wright, and S. Romanosky, CVSS: A Common Vulnerability Scoring System. National Infrastructure Advisory Council (NIAC), 2004.

[50] M. Barrett, C. Johnson, P. Mell, S. Quinn, and K. Scarfone, Guide to adopting and using the Security Content Automation Protocol (SCAP). NIST Special Publ. 800-117 (Draft), U.S. National Institute of Standards and Technology, 2009.

[51] M. Howard, J. Pincus, and J. M. Wing, Measuring Relative Attack Surfaces. Workshop on Advanced Developments in Software and Systems Security, 2003.

[52] P. K. Manadhata, D. K. Kaynar, and J. M. Wing, A Formal Model for a System's Attack Surface. Technical Report CMU-CS-07-144, July 2007.

[53] D. S. Herrmann, Complete Guide to Security and Privacy Metrics – Measuring Regulatory Compliance, Operational Resilience and ROI. Auerbach Publications, 2007, 824 p.

[54] A. Jaquith, Security Metrics: Replacing Fear, Uncertainty and Doubt. Addison-Wesley, 2007.

[55] N. Bartol, B. Bates, K. M. Goertzel, and T. Winograd, Measuring Cyber Security and Information Assurance: a State-of-the-art Report. Information Assurance Technology Analysis Center IATAC, May 2009.

[56] R. Savola, A Novel Security Metrics Taxonomy for R&D Organisations. In the Proc. of the 7th Annual Information Security South Africa (ISSA '08) Conference, Johannesburg, South Africa, July 7-9, 2008, pp. 379–390.

[57] Middleware Resource Center. Basic Message-oriented Middleware, Commercial and Open Source., Accessed May 26th, 2010 from http://www.middleware.org/mom/basicmom.html.

[58] S. A. Macskassy and F. Provost, A Brief Survey of Machine Learning Methods for Classification in Networked Data and an Application to Suspicion Scoring, Statistical Network Analysis: Models, Issues, and New Directions. LNCS Springer Berlin / Heidelberg, Vol. 4503/2007, April 12, 2008, pp. 172–175.

[59] H. Abie, Adaptive Security and Trust Management for Autonomic Message-Oriented Middleware. IEEE Symposium on Trust, Security and Privacy for Pervasive Applications (TSP) 2009, October 12-14, 2009 in Macau SAR, P.R.China.

[60] R. Savola and H. Abie, Identification of Basic Measurable Security Components for a Distributed Messaging System. The 3rd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE) 2009, June 18-23, 2009 - Athens/Vouliagmeni, Greece, IEEE Computer Society, pp. 121–128.

[61] E. Grishikashvili, Investigation into Self-Adaptive Software Agents Development. Distributed Multimedia Systems Engineering Research Group, Technical Report, Liverpool john Moors University, 27 April 2001.

[62] ITU Recommendation X.509 (08/05), Information technology – Open systems interconnection – the directory: public-key and attribute certificate frameworks, International Telecommunication Union, 2005.

[63] R. Savola and H. Abie, On-Line and Off-Line Security Measurement Framework for Mobile Ad Hoc Networks. Journal of Networks . Vol. 4, No. 7, September 2009, Academy Publisher, ISSN 1796-2056, pp. 565–579.

[64] R. Savola and H. Abie, Development of Security Metrics for a Distributed Messaging System, The 3rd Int. Conference on Application of Information and Communication Technologies, AICT2009, Azerbaijan, Baku, 14-16 October, 2009.

[65] R. Savola and P. Heinonen, Security-Measurability-Enhancing Mechanisms for a Distributed Adaptive Security Monitoring System, SECURWARE ’10, Venice, Italy, 18-25 July, 2010.

[66] IBM White Paper Autonomic Computing, an Architectural Blueprint for Autonomic Computing, Third Edition, June 2005.

[67] J. Mirkovic and P. Reiher, A Taxonomy of DDOS Attacks and DDOS Defense Mechanisms. Computer Communication Review 2004, Vol. 34; No. 2, ACM Press, USA, ISSN 0146-4833, pp. 39–54.

[68] T. Bu, S. Norden, and T. Woo, A Survivable DoS-resistant Overlay network. Computer Networks 50(9): 1281-1301 (2006)

[69] J. Wang and J. Bigham, Anomaly Detection in the Case of Message-oriented Middleware. In Proceedings of the 2008 Workshop on Middleware Security (Leuven, Belgium, December 02 - 02, 2008). MidSec '08. ACM, New York, NY, 40-42.

[70] J. DeMott, The Evolving Art of Fuzzing, 1 June, 2006, accessed May 26th, 2010, from http://www.vdalabs.com/tools/The_Evolving_Art_of_Fuzzing.pdf.

[71] J. Wang, J. Peng, J. Bigham, B. Chew, B. V. Murciano, M. Novkovic, and I. Dattani, Adding Resilience to Message-oriented Middleware. In Proc. Serene 2010 ACM, 13-16 April, 2010 London, UK.

[72] Sun Microsystems, Java Message Service API Specification v1.1. Sun Microsystems, April, 2002, accessed May 26th, 2010 from http://java.sun.com/products/jms/.

[73] TIBCO. Rendezvous. Accessed May 26th, 2010, from http://www.tibco.com/software/messaging/rendezvous/default.jsp.

[74] IBM. IBM WebSphere MQ support, accessed May 26th, 2010 from http://www-01.ibm.com/software/integration/wmq/support/.