Published in: 2013 International Conference on Availability, Reliability and Security, Regensburg, 2013, pp. 794-801. https://doi.org/10.1109/ARES.2013.106

Sink or SWIM: Information Security Requirements in the Sky

Martin Gilje Jaatun and Tor Erlend Fægri
Department of Software Engineering, Safety and Security
SINTEF ICT, Trondheim, Norway
Email: {martin.g.jaatun, tor.e.fegri}@sintef.no
http://www.sintef.org/ses

***Abstract*—Despite the inherently cooperative nature of air traffic control, the ICT infrastructure supporting it has, metaphorically speaking, largely remained isolated islands of technology. To this day, most of the interaction between ATM centers is based on voice and point-to-point data communication. Speed and accuracy of coordination is thus frequently limited by human capacities. This also imposes severe restrictions on the scale of coordination efforts among ATM centers. There are, however, changes underway. The main ambition of the System-Wide Information Management (SWIM) concept is to realize a European-wide network of interconnected ATM systems that promises, among other things, to bring substantial gains in efficiency of coordination and improved utilization of valuable airspace. This paper presents challenges, approaches and experiences from ongoing work on security requirements within SWIM.**

***Index Terms*—Security Requirements; SWIM; SESAR; ATM Security**

I. Introduction to SWIM

A key ambition of the System-Wide Information Management (SWIM) concept is to realize a European-wide network of interconnected ATM systems that promises to bring substantial gains in efficiency of coordination and improved utilization of valuable airspace.

> “SESAR will provide an effective remedy to air transport capacity bottlenecks, fills gaps in the air traffic management system, enables significant reduction of CO_2 emissions, increases safety, and reduces overall costs.” [1]

While a few fragmented solutions have been deployed, such as CFMU[1] and ACARS[2], the bulk of cooperation among ATM’s is currently based in point-to-point voice and data communication. Hence, SWIM represents a paradigm shift, as illustrated in Fig. 1 and Fig. 2.

A successful transformation of ATM systems to an interconnected system model requires the preservation of passenger safety and business continuity. Thus, both traditional resilience providing safety, and secured ATM collaboration needs to be maintained. As systems become interconnected, their safety depends critically on security. The safety-oriented mindset that has dominated aviation must now also be complemented by security[3] [2].

A. A Single European Sky

Enabling a more effective utilization of the European airspace has been a high-priority task for European Union member states for a long time. While the USA has had operational airspace management at the federal level for many years, this cross-country management level has been lacking in Europe. SES is an initiative to build a EU-coordinated airspace across Europe. In 2001, the European Commission succeeded in getting the Single European Sky (SES) legislation approved and the SES project was formally started in 2004[4]. SES aimed at giving EUROCONTROL[5] a solid framework for seamless, pan-European air traffic management. SES also includes Norway and Switzerland. Governments in Iceland and Lichtenstein are currently debating their actual involvement in SES.

> “The SES performance-driven approach focuses on the four Key Performance Areas (KPAs) of environment, cost-efficiency, safety, and capacity/quality of service.” [1]

Being a cost-driven industry, particular European-level commitments have been necessary to provide funding. SESAR is the EU’s program to implement SES. SESAR is now in its 2. phase named development. The 3. phase, deployment, is due to start in 2014. In SESAR, System Wide Information Management (SWIM) is one of the key features to accomplish SES. SWIM is the key enabler for information exchange between Air Navigation Service Providers (ANSPs), integrating air-ground and ground-ground information exchange by a comprehensive infrastructure. SWIM represents a fundamental change in the approach to information management throughout the European ATM infrastructure[6].

The main objective of SWIM is to provide trustworthy information to ATM system entities in a reliable manner. An information lifecycle perspective to information management

[1] Central Flow Management Unit, see http://www.eurocontrol.int/network-operations

[2] Aircraft Communications Addressing and Reporting System

[3] This does not imply that the *SESAR safety framework* will cover security.

[4] http://www.iata.org/pressroom/facts_figures/fact_sheets/pages/ses.aspx

[5] European Organisation for the Safety of Air Navigation

[6] http://www.eurocontrol.int/services/system-wide-information-management-swim

is at the core of SWIM. The full lifespan and the different demands for information management during the different stages of the information lifecycle will be used to determine the appropriate solution. For example, the information lifecycle perspective in SWIM implies that information processed by SWIM will be subjected to comprehensive policies regarding access, storage and archiving.

B. How, What, and Who

To ensure effective implementation and flexibility in deployment scenarios, SWIM will be based upon a publish/subscribe paradigm (PSP.) Using PSP, information producers and information consumers are loosely connected. PSP also has important consequences for the security of the system as a whole. Security in large-scale distributed systems is always hard, and this is made even more challenging due to the hitherto weak requirements towards information security in the existing systems.

Furthermore, SWIM must embrace a significant diversity in the information to be exchanged:

- Aeronautical – Information resulting from the assembly, analysis and formatting of aeronautical data.
- Flight trajectory – the detailed route of the aircraft defined in four dimensions (4D), so that the position of the aircraft is also defined with respect to the time component.
- Aerodrome operations – the status of different aspects of the airport, including approaches, runways, taxiways, gate and aircraft turn-around information.
- Meteorological – the information on the past, current and future state of the earth's atmosphere relevant for air traffic.
- Air traffic flow – the network management information necessary to understand the overall air traffic and air traffic services situation.
- Capacity and demand – information on the airspace users' needs of services, access to airspace and airports and the aircraft already using these resources.

A notable exception is surveillance data. Positioning information from radar, satellite navigation systems, aircraft data-links, etc. is not exchanged through SWIM. Radar data will remain exchanged through dedicated network links from radars to Air Traffic Control (ATC) centers, and the position of aircraft will be exchanged through for instance the ADS-C protocol, which is not part of SWIM. The most critical information from ground to aircraft (the clearances) will still be exchanged through existing data-link protocols (Controller Pilot Data Link Communications (CPDLC)[7], . . .), which are not part of SWIM.

In addition, SWIM enables a wide range of actors to share information – reliably and with confidence – through different applications. These actors include:

- Pilots – taking off, navigating and landing the aircraft;
- Airport Operations Centres – managing departures, surface movements, gates and arrivals;
- Airline Operations Centres - building schedules, planning flight routings and fuel uplift, ensuring passenger connections and minimizing the impact of delays;
- Air Navigation Service Providers (ANSPs) – organizing and managing the airspace over a country and with Air Traffic Services – managing air traffic passing through their airspace;
- Meteorology Service Providers – providing weather reports and forecasts;
- Military Operations Centres – planning missions, blocking airspace to conduct training operations, fulfilling national security tasks.

SWIM seeks to span the majority[8] of ATM systems, data domains, and aviation processes (such as planning, execution and post-execution.) Given the wide range of ATM stakeholders, it is not expected that one solution and certainly not one single technology will accommodate all needs. Furthermore, the cost-driven nature of ATM advances require that SWIM can be adapted to the economic limitations of different business solutions and operational activity. Nevertheless, it is recognized that global interoperability and standardization are essential, and SWIM is expected to be an important driver for new and updated standards.

The benefits of SWIM can be illustrated by a couple of examples. In our first example, a sudden severe meteorological event causes the closure of an airport for several hours. This information is communicated via SWIM and enables fast and well-informed re-routing of active flights. Planned flights are re-planned.

In our second example, Flight Plan data are shared between the ATC centers that will control the aircraft along its trajectories. These centers may ask for modifications of the planned trajectory to the center controlling the aircraft. This enables optimization of trajectories across ATC boundaries.

C. System context

SWIM is a distributed information processing system that consists of a number of components that exchange information. The paradigm shift represented by SWIM is illustrated in Fig. 1 and Fig. 2 [3].

SWIM, once deployed, will consist of a set of nodes that are interlinked with an underlying network. The SWIM nodes house and execute what is called The SWIM Technical Infrastructure (SWIM TI). The main objective of the SWIM TI is to facilitate the operational activities within the system. Internally, SWIM TI is structured into so-called Functional Blocks that consist of groups of functions that address specific needs for the operational activity in question. In deployment, SWIM TI is run on multiple nodes within the system, and each SWIM TI serves as a (local) access point for the various operational activities supported within the ATM System. The SWIM entities are connected with each other to exchange

[7] http://www.eurocontrol.int/category/keywords/cpdlc

[8] with the already mentioned exception of surveillance

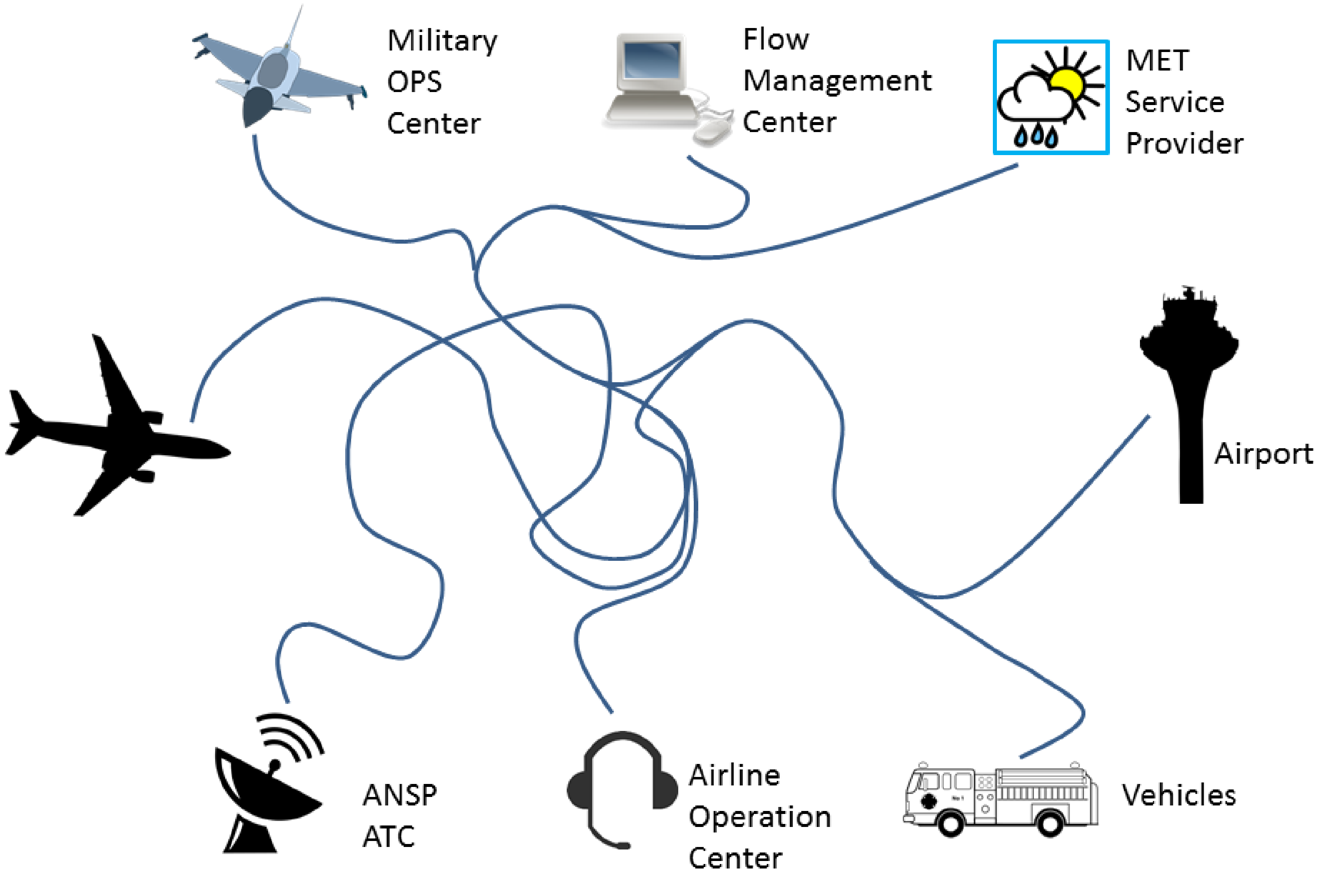

Fig. 1: Aviation communication before SWIM (based on SESAR SWIM factsheet [3])

data amongst different ATM environments. A common SWIM Infrastructure is used that provides technical services – e.g. for access control and registry – in a centralized way. The SWIM Data Services facilitate exchange of data between different SWIM entities.

II. Security challenges in SWIM

The aviation domain is recognized for being a driver for safety-critical engineering [4]. Safety standards, formal methods, n-version programming, and safety cases have seen widespread use in software development for aviation systems.

In the quest for increased efficiency in airspace utilization, the aviation business is opting for an interconnected system model. [9]. This transition has a range of fundamental implications for the proper handling of security:

1) Due to enormous impacts of potential mishaps, safety and security become ’top-priority’ requirements;
2) The number of stakeholders and individual roles is immense. This is matched by a similar diversity in information needs and security requirements;
3) A large number of existing applications and systems must be supported. Security needs to be maintained across each application and system;
4) Security is a pervasive quality that demands holistic thinking – both technical and social systems (organizational and human aspects) must be considered as equal parts of the whole.

Security is needed both for ATM resilience (that is, for safety and for business continuity) and collaborative support[10] ; this paper focuses on the former aspect.

SWIM adopts the following ambition for security:

> “To fully protect the information during its lifetime, each component of the information processing

[9] The manufacturing industry, however, is not advocating a component-based model, and we therefore cannot expect to see ”plug-and-play” SWIM solutions in the near future, if ever.

[10] Collaborative support is defined in SESAR as: *“providing services or information from ATM to another agent such as law enforcement, military agencies, emergency services or incident investigation agencies relating to an act of unlawful interference”* [5]

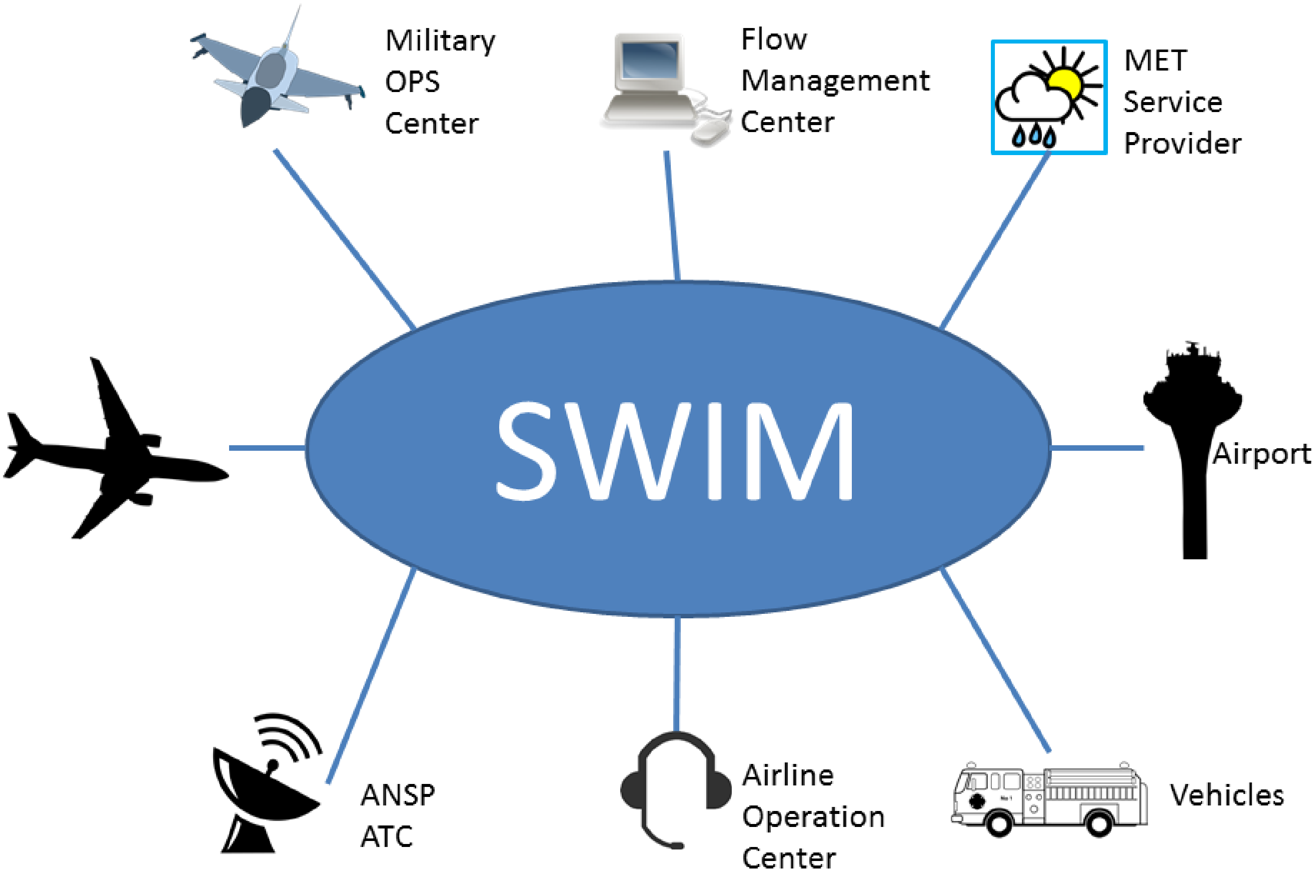

Fig. 2: Aviation communication after SWIM (based on SESAR SWIM factsheet [3])

> system must have its own protection mechanisms, by building up, layering on and overlapping of security measures through a so-called defense in depth mechanism. The main layers of intervention are at network and data level." [6]

Legislation for ATM security is being established[11], with ICAO ANNEX 17 and in Europe CIR 1035/2011 [7]:

> "Air navigation service providers shall establish a security management system to ensure:
> (a) the security of their facilities and personnel so as to prevent unlawful interference with the provision of air navigation services;
> (b) the security of operational data they receive or produce or otherwise employ, so that access to it is restricted only to those authorized." [7]

The transition to SWIM is a real challenge for security, since it implies:

- going from communication mostly based on point-to-point exchanges to net-centric exchanges, with an increased number of exchanges and thus an increased attack surface;
- using COTS and open source software, which may have widely known lists of vulnerabilities;
- complying with national regulations;
- complying with the security requirements for interoperability with US SWIM.

III. Safety versus Security

The terms *Safety* and *Security* refer to related, but quite different concepts. The inability of the system to affect its environment in an undesirable way is usually called safety [8]. ICAO defines safety as

> "[...] the state in which the risk of harm to persons or of property damage is reduced to, and maintained at or below, an acceptable level through a continuing process of hazard identification and risk management." [9]

[11] Note that this is not an exhaustive list of applicable legislation [6].

Safety incidents are typically associated with some sort of malfunction which can be triggered by, e.g., material fatigue, component failure or extreme weather.

Security, on the other hand, can be understood as the inability of the environment to affect the system in an undesirable way [8]. A secure system is able to protect itself and its related assets despite ongoings in its environment. For example, the ultimate goal of a secure system is to withstand attacks – malicious acts of people. In SESAR, information security is defined as

> "[. . .] protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction." [6]

While a safety analysis will try to calculate, e.g., a system-wide probability of failure, it is not possible to calculate a probability of attack in the same manner. Consequently, it is not possible to calculate a "Mean Time To Attack". Furthermore, the likelihood of an attack can change over time, as the source and nature of threats may change.

On the other hand, there are similarities for instance when performing risk assessment [10] (even though risk assessments for safety are not the same as security risk assessments). Furthermore, the results of a safety assessment (failure mode analysis) may be used as inputs to the security risk assessment. This is particular true when assessing impact of incidents. There are thus synergies between safety and security that hopefully can be exploited in future work. However, it must be recognized that the relation is not necessarily bilateral; the safety work that is being performed in SESAR does not comprise security aspects.

In safety-critical environments such as aviation, it is important to acknowledge that many security attacks can have safety consequences. There are numerous examples of this in popular culture [11], but in addition to blatant attacks to make airplanes crash directly, one could also imagine more subtle attacks to cancel out safety mechanisms; the latter variety might not even be recognized as attacks, but attributed to "freak accidents". This could in turn give the attackers a feeling of impunity, enabling them to conduct further attacks without detection.

IV. ISO/IEC 25010 FOR SECURITY – CONCEPTUAL PROBLEMS

The ISO/IEC 25010 standard is a conceptual framework for classification and evaluation of quality attributes. ISO 25010 seeks to bring amends to limitations in the former standard – the ISO 9126 [12]. A fundamental problem with ISO 9126 was the ambiguity among classifiers [13]. Ambiguity reduces the usefulness of standards and will ultimately undermine their importance in the field. However, initial experiences with the ISO 25010 gives rise to concern that ambiguity will still be problematic.

The security attributes in ISO 25010 are: confidentiality, integrity, non-repudiation, accountability and authenticity.

ISO 25010 defines integrity as "degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data." The definition of integrity has a significant overlap with confidentiality. Access to information is part of confidentiality – i.e. keeping information confidential.

Non-repudiation and accountability are attributes at separate conceptual levels and are therefore likely to create more confusion than understanding and clarity. If an actor is held accountable for an action, it is already assumed that the actor cannot deny (or repudiate) involvement.

In fact, the two attributes should not be separated, and non-repudiation would be part of accountability. Technically, non-repudiation is provided through the use of digital signatures, and is thus inextricably linked with public-key cryptography. It is our experience that the technical concept of non-repudiation is poorly understood, and is thus perhaps not suited for high-level requirements intended to be used in discussions and prioritization work by stakeholders.

In terms of security in the digital domain, authenticity is largely irrelevant. In the normal usage of the term, an object is authentic if it is 'the original physical artifact' and not a copy. Whereas an authentic object in the physical world represents the true original, 'the original' is not meaningful term in the digital domain. An actor can never be sure that it is looking at the 'true original' or a copy. In fact, in a distributed system, an actor cannot reasonably expect or assume that only one version of an object exist. In the digital domain, any object can be, and will easily be, copied. It is simply a matter of bit-wise reproduction. An object being copied or accessed over a network is transmitted as a string of bits. However, where authenticity gives meaning in the digital domain of a distributed system such as SWIM is an actor's attempt to determine whether *the copy* of a particular object *is not altered* on its way to the actor. The copy cannot be any less authentic than the original unless it has been changed. If it has been changed during the copy, its *integrity* is violated. This is the definition of integrity. Note also that *message integrity* typically also adds a verification that the message originates from the claimed sender (i.e., is authentic). Hence, we have to suggest that the security term authenticity be deemed irrelevant as its intent is covered by integrity.

Availability is in ISO 25010 classified as a reliability attribute. This is problematic. Because Denial of Service (DoS) attacks comprise a large percentage of the total attacks in distributed systems, and thus clearly a major security issue. Granted, loss of availability can also have natural causes, but is is important to retain this as a security attribute in order not to lose sight of the security implications. Furthermore, increased availability from a safety perspective may in some cases lead to a reduction of security (e.g., if network duplication introduces new security vulnerabilities).

V. REQUIREMENTS

The SWIM security requirements process has received input from many sources, including predecessor project such as SWIM-SUIT [14], the ISO/IEC 27002 standard [15] and input from various aviation-specific standards and guidelines.

Furthermore, after the first iteration of high-level security requirements was created, an initial security risk assessment (SRA) [10] of SWIM security solutions was performed; this SRA identified several new security requirements for the next iteration.

A. MSSC

It has been proposed that security should be considered in all part of the development process, not only in the projects that have "security" in their name. A Minimum Set of Security Controls (MSSC) has therefore been drafted to ensure that a minimum security baseline can be achieved in every project. The intention is that every SESAR project should either ensure that they comply with the MSSC, or provide a compelling argument for why a given control in the MSSC is not relevant for this project. Furthermore, it is the spirit rather than the letter of the MSSC that is important, so it will be totally acceptable if it can be shown that one or several different controls cover the intention of a given control in the MSSC.

The approach of having a baseline set of requirements is also known from other critical infrastructure settings, e.g., the Norwegian Oil and Gas Association have published such requirements that are expected to be adhered to by all actors on the Norwegian continental shelf [16]. The baseline approach is useful not only because it advocates a minimum security level, but because it requires every project developing a piece of the total ATM solution to think about security.

B. Security requirement categories

Based on the prior comments, we have proposed the following list of security requirement categories as a first iteration:

- Confidentiality: The degree to which a product or system ensures that data are accessible only to those authorized to have access.
- Integrity: The degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.
- Availability: The degree to which a system, product or component is operational and accessible when required for use.
- Non-repudiation: The degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later.
- Accountability: The degree to which the actions of an entity can be traced uniquely to the entity.
- Security governance: A category of security requirements that are not directly part of the technical infrastructure, but will be important for ensuring that the infrastructure remains properly managed.
- Implementation guidelines: A category for security requirements that pertains to the recommended practices used during actual implementation of the SWIM TI.

Essentially, these categories should be considered our quality attributes. As the quality attributes are at a high level of abstraction and have been thoroughly reviewed, they are more likely to ensure good coverage for the security requirements. This is because we can easily identify those categories with few or missing requirements. This would be more difficult if requirements were targeting technical solutions directly. The template to specify requirements is illustrated in Table I.

C. Security requirement examples

Space does not allow documenting all SWIM security requirements in this paper, but in the following we will provide examples that illustrate the level and scope.

1) Confidentiality example: **Every information exchange according to SWIM data patterns from the trusted network or the public network (a.k.a. Internet) shall be subject to authorization.**

Rationale: Confidentiality depends on accesses to information being controlled by authorization. Network components should re-authorize publishers at each entry to the network. Due to the highly dynamic environment every transaction should be re-authorized even it is coming from the protected network or the outside network because some publishers can belong to more than one domain where each has different security properties.

2) Integrity example: **SWIM-nodes and/or ATSUs SHALL issue return receipts upon service update requests**

Rationale: Implementation of return receipt counters the vulnerability: "there is no end-to-end check that messages are sent or received on the network" and contributes to reduce the risk of modification of the FO database through man-in-the-middle attacks on the network.

3) Availability example: **SWIM infrastructure SHALL formulate run-time service security alerts to be provided to the SWIM supervision body.**

Rationale: Attacks to jeopardize availability can become known more quickly with the use of alerts. Timely reaction to incidents can reduce consequences severely.

VI. Discussion

It has been argued that security should be considered a quality attribute of software systems similar to other qualities, such as usability, correctness, performance and maintainability [17]. With a carefully constructed framework for quality attributes, there need not be distinctions between functional and non-functional quality attributes [18].

So far, our work on security requirements in SWIM has suggested that a quality attribute based approach provides a useful structuring of demands towards security. The quality attributes are abstract and sufficiently high-level to warrant coverage. It would have been very easy to forget certain quality attributes if the focus was initially on technical solutions.

It is important to note that this is true not only for components with security functions, but also for any other component that is exposed to external input. Various attacks on PDF readers in the past years are a testament to this [19]. Thus, developers should focus on developing secure features, also when these are not *security* features. Another important insight is that security needs to be built into systems from the very

Identifier	Unique reference that contains security requirement category and a sequence number
Requirement	The requirement description
Title	An abbreviated requirement title
Status	“In progress” or “complete”
Rationale	Justification of why the requirement is included.
Covering MSSC §	What, if any, objective of the MSSC the requirement is covering.
Source	Where, if applicable, the requirement originated.

TABLE I: Requirement structure

start of development – it seldom works well to bolt it on after-the-fact. Unfortunately, there are no reliable metrics that can measure the level of security of a given piece of software [20], and thus the only viable approach is to ensure that all phases of system development take security into account.

Accommodating security requirements is, however, just as demanding as other types of requirements. Security is a cross-cutting concern that is enabled or inhibited by a multitude of design decisions, at different levels in the design. Hence, security also needs to be part of the mindset from the beginning where architectural decisions are made [21].

The danger lies in assuming that if security is a quality attribute, expertise in quality assurance is all you need to make secure systems. As in any engineering discipline, specialist knowledge is required to ensure a good result. There are plenty of examples in the past where poor security solutions have been rolled out, despite plenty of expert knowledge have been available to offer advice to the contrary [22], [23] [12].

VII. Conclusion

In this paper we have outlined how security requirements are being addressed within SWIM. We have discussed challenges related to working with security in a safety oriented domain, and pointed out that there is a difference between treating security as a quality attribute and saying that specialist security knowledge is not required.

Good security requirements can only result from a combination of security and domain knowledge, and in our case, the domain also presupposes safety expertise.

Acknowledgements

This paper is the result of collaborative research co-funded by the SESAR Joint Undertaking. However, the paper does not purport to represent the views of SESAR JU or any SESAR partner.

References

[1] (2012) European atm master plan. [Online]. Available: https://www.atmmasterplan.eu/

[2] H. Chivers and J. Hird, “Security Blind Spots in the ATM Safety Culture,” in *Proceedings of the International Workshop on Security of ATM and other Critical Infrastructures*, 2013.

[3] (2011) SESAR Factsheet: System Wide Information Management (SWIM). [Online]. Available: http://www.sesarju.eu/sites/default/files/documents/reports/factsheet-swim.pdf

[4] I. A. Herrera, A. O. Nordskag, G. Myhre, and K. Halvorsen, “Aviation safety and maintenance under major organizational changes, investigating non-existing accidents,” *Accident Analysis & Prevention*, vol. 41, no. 6, pp. 1155 – 1163, 2009, accident Modelling and Prevention at ESREL 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001457508001012

[5] J. Hird and C. Machin, “SESAR ATM Security Reference Material - Level 2 - 2013,” 2013, 16.06.02 Deliverable D101.

[6] I. Lesot et al., “SWIM Security requirements and design for iteration 2.1,” 2013, 14.02.02 Deliverable D21.

[7] “COMMISSION IMPLEMENTING REGULATION (EU) No 1035/2011 laying down common requirements for the provision of air navigation services and amending Regulations (EC) No 482/2008 and (EU) No 691/2010,” 2011, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?-uri=OJ:L:2011:271:0023:0041:EN:PDF.

[8] M. Line, O. Nordland, L. Røstad, and I. A. Tøndel, “Safety vs. Security?” in *Proceedings from Probabilistic Safety Assessment and Management (PSAM)*, 2006.

[9] “ICAO Safety Management Manual (SMM),” 2006, doc 9859 AN/460.

[10] J. Touzeau, S. Chopart, E. Hamon, M. S. Cidoncha, J. Hird, K. Lamont, T. Niclasen, and B. Solhaug, “SESAR ATM Security Risk Assessment Method,” 2013, 16.02.03 Deliverable D02.

[11] R. Harlin (dir.), W. Wager (Writ.), and B. Willis (Perf.), “Die Hard 2,” Film, 1990, Twentieth Century Fox Film Corporation.

[12] *Information Technology - Software Product Quality. Part 1: Quality Model*, ISO/IEC Std. 9126, 1991.

[13] H.-W. Jung, S.-G. Kim, and C.-S. Chung, “Measuring software product quality: A survey of ISO/IEC 9126,” *IEEE Software*, vol. 21, no. 5, pp. 88–92, 2004.

[14] P. Fantappié, D. Smith, N. Silvester-Thorne, G. Bogdos, B. Badanik, A. Kazda, K. Havel, D. Scarlatti, and R. Schreiner, “Swim-suit security requirements,” 2008. [Online]. Available: http://www.swim-suit.aero/swimsuit/projdoc2.php?action=download&id=56

[15] *Information technology - Security techniques - Code of practice for information security management*, ISO/IEC Std. 27002, Std., 2005.

[16] “Norwegian Oil and Gas Association recommended guidelines for Information Security – Baseline Requirements for Process Control, Safety and Support ICT Systems,” Norwegian Oil and Gas Association, Tech. Rep. OLF Guideline 104, 2009. [Online]. Available: http://www.norskoljeoggass.no/no/Publikasjoner/Retningslinjer/Integrerte-operasjoner/104-Anbefalte-retningslinjer-krav-til-informasjonssikkerhetsniva-i-IKT-baserte-prosesskontroll–sikkerhets–og-stottesystemer/

[17] M. G. Jaatun and P. H. Meland, “Guest Editorial Preface: Special Issue from the 5th International Workshop on Secure Software Engineering,” *International Journal of Secure Software Engineering*, vol. 2, no. 4, pp. i–ii, 2011. [Online]. Available: http://www.igi-global.com/Files/Ancillary/1947-3036_2_4_Preface.pdf

[18] L. Bass, P. Clement, and M. Klein, *Software Architecture in Practice*, 2nd ed. Addison-Wesley, 2003.

[19] N. Villeneuve. (2013) Malicious PDFs On The Rise. [Online]. Available: http://blog.trendmicro.com/trendlabs-security-intelligence/malicious-pdfs-on-the-rise/

[20] M. G. Jaatun, “Hunting for Aardvarks: Can Software Security be Measured?” in *”Proceedings of the International Cross Domain Conference and Workshop (CD-ARES)”*, 2012, pp. 85–92. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-32498-7_7

[21] T. E. Fægri and S. Hallsteinsen, “A Software Product Line Reference Architecture for Security,” in *Software Product Lines*, T. Käkölä and

[12] Admittedly, there have been cases in the past where specifically cryptographic systems have been intentionally weakened for political reasons, but this is no excuse for poor design.

J. Duenas, Eds. Springer Berlin Heidelberg, 2006, pp. 275–326. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-33253-4_8

[22] S. Somogyi. (2001) 802.11 and swiss cheese. [Online]. Available: http://www.zdnet.com/news/802-11-and-swiss-cheese/115357

[23] F. A. Stevenson. (1999) Cryptanalysis of contents scrambling system. Archived web page. [Online]. Available: http://www.cs.cmu.edu/d̃st/DeCSS/FrankStevenson/analysis.html