

Resilience in the Cyberworld

Subjects: [Computer Science, Theory & Methods](#) | [Computer Science, Software Engineering](#) | [Computer Science, Artificial Intelligence](#)

Contributor: Elisabeth Vogel

Resilience is a feature that is gaining more and more attention in computer science and computer engineering. However, the definition of resilience for the cyber landscape, especially embedded systems, is not yet clear.

cyber-resilience

security

redundancy

resilience engineering

1. Introduction

The cyberlandscape of the 21st century is constantly growing and becoming increasingly complex, covering areas such as telemedicine, autonomous driving, etc. Our societies, as well as individuals, are highly dependent on these systems working correctly 24/7. In order to be able to cope with the increasing complexity and the unprecedented importance of cybersystems, new and innovative methods and technologies have to be applied. The concept of resilience is receiving increasing attention in this respect, which is reflected above all by the steadily growing number of publications on the topic. **Figure 1** shows how the number of publications has increased since 2012. The diagram in **Figure 1** shows only publications with the keyword *Cyber-Resilience*. Beneath its attention in science, the concept of resilience has already reached industry. US-American streaming provider Netflix is considered a pioneer in the application of resilience in the form of highly redundant infrastructure. However, the principles of resilience are not only found in the hardware components of Netflix. The software architecture also demonstrates the application of various methods to increase resilience. The example of Netflix shows how important resilience becomes with increasing complexity ^[1].

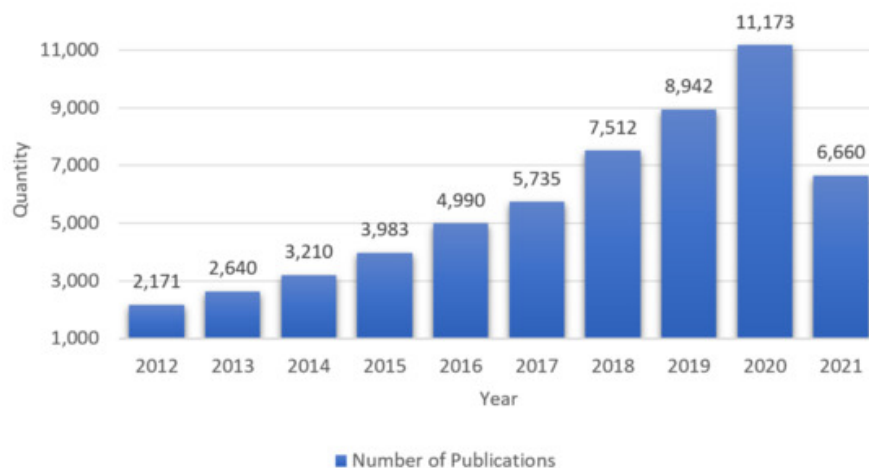


Figure 1. Number of publications with the keyword Cyber-Resilience from 2012 to 2021 (July) ^[2].

However, the term “resilience” is used in many ways in IT. In some cases, resilience is described as “extreme reliability” ^[3] or used as a synonym for fault tolerance ^{[4][5]}. In ^[6] it is described that resilience is fault tolerance with the key attribute robustness. Anderson, in ^[6], extended the definition of fault tolerance by the property robustness, and called the new definition resilience. In recent publications, however, resilience is defined several times as an independent term ^{[7][8]}. In publications ^[9] and ^[10], some aspects are added to the concepts already applied in ^[7]. In ^[9], the model of ^[7] is extended. In ^[10], methods used to achieve cyber-resilience are listed as possible measures against cyberattacks (keyword, IT security).

2. Definitions

In the literature of recent years, there have been many definitions of resilience, some of which differ considerably. As described in [4], the content of the definitions strongly depends on the respective fields of application.

Resilience is derived from the Latin *resilire*, and can be translated as “bouncing back” or “bouncing off”. In essence, the term is used to describe a particular form of resistance.

How the term resilience is used in different other disciplines (material science, engineering, psychology, ecology) is described in [11].

Additionally, in computer science, the term resilience has been defined several times from different points of view. As described in [4] for example, resilience is often used as a synonym for fault tolerance. However, recent publications show that this approach has been replaced by the view that resilience is much more than fault tolerance (see [8]).

One of the first definitions was presented 1976 in [3], and describes the concept of resilience as follows:

“He (remark: the user) should be able to assume that the system will make a “best-effort” to continue service in the event that perfect service cannot be supported; and that the system will not fall apart when he does something he is not supposed to”.

In addition, [3] mentions attributes that constitute resilience as part of its definition. The attributes are the following: **error detection**, **reliability**, **development capability** and **protection against misuse** in the sense that the misuse of a system by individual users has only negligible effects on other users. According to Alsberg [3], these four attributes of a resilient system can be summarized as the attempt to describe extreme reliability and serviceability. In summary, a partial failure of a system should not have any effect on an individual user, so the system can be assumed to be highly reliable. Should nevertheless a partial failure or a defect occur, the best possible continuation of the services provided should be guaranteed. In extreme cases, this continuation can also be achieved by performing graceful degradation of services.

The approach of continuing a service of a system even under transient effects, permanent load or failures is also described in [12].

“A resilient system keeps processing transactions, even when there are transient impulses, persistent stresses, or component failures disrupting normal processing. This is what most people mean when they just say stability. It's not just that your individual servers or applications stay up and running but rather that the user can still get work done”.

According to Nygard [12], a system must remain stable in case of tensions or stress situations or failures. As a consequence, involved (sub-) systems, or possibly also users, can still continue their work. The system must also be able to continue fulfilling at least its rudimentary functions despite any restrictions that may occur. The scope of these rudimentary functions may have been defined as part of the Risk Management, for example. Risk management also shows at what level of functional loss the entire system function according to its specifications can no longer be provided.

Laprie [13] proposes two definitions of resilience. The first definition is as follows:

“The persistence of service delivery that can justifiably be trusted, when facing changes”.

According to Laprie, this definition corresponds in principle to the original definition of reliability. In a second definition, Laprie offers an alternative which provides a more detailed description:

“The persistence of the avoidance of failures that are unacceptably frequent to severe, when facing changes”.

In [13], resilience is described as the persistence of service delivery when changes occur that have system-wide effects. These changes can be functional, environmental or technological. In addition, changes can either be planned (for example: initialized by an update), the timing of their occurrence can be unpredictable, or they can be completely unexpected. The duration of changes is also taken into account: short-term, medium-term, long-term. This refers to the duration of the impact of the change on the system or a subsystem.

In the collection of papers from 1985 [6], robustness was already mentioned in connection with resilience. About 30 years later, in [14], this connection is concretized. Resilience is defined as the trustworthiness of a software system to adapt to adverse conditions. The software system should accept and tolerate the consequences of failures, attacks and changes inside and outside the system boundaries. This is defined as an approach for robustness:

“Software resilience refers to the robustness of the software infrastructure and may be defined as the trustworthiness of a software system to adapt itself so as to absorb and tolerate the consequences of failures, attacks, and changes within and without the system boundaries”.

The definition of resilience was further specified in [14]. Florio [14] refers to the definition already mentioned in [4] and another definition in [15]. This definition states that resilience can be characterized as a measure of the persistence of both functional and non-functional features of a system under certain and unpredictable disturbances. After analyzing these two definitions, according to Florio, resilience is the ability to act and balance between two main behaviors:

- (1) Continuous readjustment with the aim of improving the fit of the systems' environment, and compensating for both foreseeable and unforeseeable changes in the system environment.
- (2) Ensure that the said changes and adjustments from 1) do not affect the identity of the system. This means that its specific and distinctive functional and non-functional features should not be affected.

Ref. [7] deals with the management of water resources, and was written from the perspective of the USACE (US Army Corps of Engineers). According to Rosati [7], resilience is a cycle consisting of anticipation, resistance, recovery and adaptation. Anticipation is the starting point of the cycle, while adaptation marks the end. The cycle is started by the occurrence of an event that affects the system in some way. This event is called a disruption. Specifically, Rosati defines resilience (in this case coastal resilience) as follows:

“(Coastal) resilience is defined as the ability of a system to prepare, resist, recover, and adapt to disturbances in order to achieve successful functioning through time”.

A disturbance occurs here as an effect of a hazard on the infrastructure, system, etc. A hazard is an environmental or adverse anthropogenic condition.

The article by Clark-Ginsberg published in 2016 [16] defines the ability of a system to reduce the extent and the duration of disruptions as resilience. Disruptive events are not always predictable, but when they occur they are supposed to lead to a learning and adaptation effect of the system. Adaptation is crucial when it comes to realizing resilience against cyberaccidents, since the cyberlandscape is developing very rapidly. Clark-Ginsberg says in his article that errors must be detected and understood. It must be possible for the system to adapt to the errors or the error situation and a fast recovery must be guaranteed. The system must recover quickly after the occurrence of an error. If this is not possible, the error and the resulting faulty system environment must be dealt with appropriately.

The U.S. National Institute of Standards and Technology (NIST) [8] coins the term cyber-resilience to clearly distinguish its approach from the general definitions of resilience. Cyber-resilience is the following property:

































“Cyber Resilience is defined in this publication as the ability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises on systems that include cyber resources”.






According to NIST, the definition of cyber-resilience refers specifically to all entities that contain cyber-resources. A cyber-resource is an information resource that creates, stores, processes, manages, transmits, or disposes information in electronic form and that can be accessed over a network or by network methods. The definition of NIST can therefore be applied to a system, a mechanism, a component or a system element, a common service, an infrastructure or a system of systems, etc.

Publications [9] and [10] do not present completely new definitions for cyber-resilience. These publications refer to already-known sources such as NIST [8] for a definition. Publication [9] by Carías extends the previously mentioned circuit (Rosati [2]) to include the ability of a system to detect threats. Carías calls this new cycle the cyber-resilience life cycle. Publication [10] by Hopkins considers cyber-resilience as an effective countermeasure against cyberattacks or cyberthreats. In [17], the link between security and cyber-resilience is also discussed.

The publications selected here show that the type and scope of the definitions of resilience depend very much on the respective (informatics) application area. However, some key actions can be filtered out, which appear at least partially in all the publications considered here: Anticipating, resisting, recovering, detecting and adapting (to threats of any kind). In some publications, such as [7][8][18] these key actions are even mentioned explicitly. Thus, we filtered these five key actions out of the definitions. **Table 1** shows which key action is mentioned in which of the publications considered here. In addition, **Table 1** also clearly indicates that the definition of resilience has been becoming more and more complex over the last decades; the number of key action mentions per publication has increased from two to all five in the most recent publications. Each of the five key actions can be assigned different attributes and behaviors. They are described in the following section.

Table 1. Sources mentioning key actions.

No.	Publication	Key action				
1	Alsberg, 1976					
2	Nygard, 2007					
3	Laprie, 2008					
4	Hollnagel, 2011					
5	Florio, 2013					
6	Rosati, 2015					
7	Clark-Ginsberg, 2016					
8	NIST, 2018					
9	Carías, 2020					
10	Hopkins, 2020					

 **Anticipation**
 **Resistance**
 **Recovery**
 **Adaptation**
 **Detection**

3. Key Actions and Attributes

[Section 2](#) introduced the key actions anticipation, resistance, recovery, detecting and adaptation. These five key actions can be defined as follows. Comparable definitions are listed in [7][9][10][19], for example:

- **Anticipation:** Anticipation is a process that enables the system to prepare for a disruption or an attack that may occur.
- **Resistance:** Resistance is the ability to withstand the effects of a disruption or an attack and maintain a certain level of functionality.
- **Recovery:** The system must be able to recover the lost functionality.
- **Adaptation:** The system must be able to put itself in a state by responding more efficiently to the disturbance in the future. This essentially means that the system will lose less functionality in the future and the recovery time will also be less.
- **Detection:** Detection means that a system can detect a disturbance or an attack in order to initiate appropriate countermeasures.

Each key action comprises several attributes (In the publications presented here, the terms attribute, feature and measure were used synonymously. For the sake of clarity, only the term attribute will be used in this article, representing Feature and Measure). These attributes can be derived directly from the definitions or were explicitly mentioned in the publications. The attached descriptions of the attributes can be found in this way in [\[12\]](#)[\[20\]](#), for example:

- **Robustness:** Still function reliably under adverse conditions.
- **Reliability:** Continuity of service.
- **Availability:** Readiness for usage.
- **Evolvability:** Ability to accommodate changes.
- **Adaptability:** Ability to anticipate to changes.
- **Security:** Crime Prevention.
- **Safety:** Accident prevention.
- **Assessability:** Check for correctness of data (plausibility check).
- **Integrity:** Nonoccurrence of incorrect system alterations.

Ref. [\[3\]](#) describes the four main attributes of a resilient service. First, a resilient service must be able to detect and correct errors. Further, the resilient service must be so robust and reliable that a user expects the service not to fail. If the service is capable of always detecting n errors and recovering from those errors, the $n + 1$ error is not catastrophic. This only applies under the condition that the system offers perfect detection and recovery of n errors. The resilient service is therefore able to anticipate the $(n + 1)$ th error in such a way that its negative consequences for the service can be minimized. This corresponds to a simple definition of evolvability. As a fourth key attribute, Alsberg [\[3\]](#) cites the ability of a resilient service to tolerate abuse by a single user in such a way that this abuse has negligible impact on the other users of the service. Alsberg does not specify misuse, but if a malicious and intentional action is assumed, then this misuse protection corresponds to the security feature of availability. Alsberg summarizes the following features: robustness, reliability, evolvability and security.

Ref. [\[12\]](#) claims that robustness under all conditions is the most important property of a resilient system. According to Nygard [\[12\]](#), this robustness is directly related to the reliability of a resilient system. This connection is obvious, since a system that is not stable cannot be reliable either.

This understanding of robustness and reliability is also illustrated in [13]. Assessability is also an important property, because a resilient system must be able to validate the correctness or plausibility of sensor data, for example. Laprie [13] also mentions diversity as another important basic property. Diversity can be understood here as a basic idea of redundancy, because according to Laprie, diversity in a system (of hardware components, for example) should prevent the occurrence of single point of failure.

In [21], it is also described that reliability is a key feature of resilience. The ability to detect a fault before it occurs is also essential. However, this only refers to faults that can be anticipated on the basis of existing information. A resilient system must be able to minimize the negative effects of a disturbance by anticipating it. This is achieved by constantly updating information about the disturbances that have already occurred and been treated. This process can be understood as the ability to evolve.

According to [14], the following attributes are essential for a resilient system: reliability, evolvability and integrity. Reliability and evolvability are related to resilience, as described in the previous definitions. Integrity, according to Florio [14], means that a resilient system does not lose its intention after adaptation or application of changes regarding a failure. This refers mainly to its functional and nonfunctional characteristics.

In the publications [7][8][16], the following abilities: anticipate, resist, recover and adapt, are directly mentioned as the four basic attributes, or, as in NIST, the four basic goals of resilience. Publications [9] and [10] extend this approach to include the ability to detect a threat. Detection is important for the self-assessment of a system, because if a threat cannot be detected, the system cannot initiate appropriate countermeasures

Discussion of the Relationships between Key Actions and Attributes

The attributes are directly related to the key actions. This relationship becomes important when it comes to the concrete design or practical modeling of cyber-resilient systems. In order to design a system to be cyber-resilient or to make an existing system cyber-resilient, it must first be examined how, for example, the ability to resist can be achieved or improved. At this point, attributes must be considered. For example, to make a system resistant, methods can be used that increase robustness. Methods of reliability and security are also helpful here. It is to be considered here that the use of several different methods can lead to dependencies, which must be considered. For example, a method that makes a system more adaptable could at the same time reduce its security. **Table 2** shows the mapping of the mentioned attributes to the key actions according to our understanding.

Table 2. Mapping of the mentioned attributes to the key actions.

	Anticipation	Resistance	Recovery	Adaptation	Detection
Robustness		X	X		
Reliability		X	X		
Adaptability		X	X	X	
Evolvability				X	
Security	X	X			X
Safety	X	X			X
Assessability					X
Integrity	X	X			

The various key actions have interdependencies, which need to be taken into account when building a theoretical model of cyber-resilience. In the following section, such models [7][9][10] are briefly presented.

References

1. Tseitlin, A. Resiliency through Failure: Netflix's Approach to Extreme Availability in the Cloud; 2013. Available online: <https://qconnewyork.com/ny2013/node/281.html> (accessed on 15 November 2021).
2. Web of Science. Available online: <https://clarivate.com/webofsciencigroup/solutions/web-of-science/> (accessed on 20 July 2021).
3. Alsberg, P.A.; Day, J.D. A Principle for Resilient Sharing of Distributed Resources. In Proceedings of the 2nd International Conference on Software Engineering October, San Francisco, CA, USA, 13–15 October 1976; pp. 562–570.
4. Meyer, J.F. Defining and evaluating resilience: A performability perspective. In Proceedings of the International workshop on performability modeling of computer and communication systems (PMCCS), Eger, Hungary, 17–18 September 2009; Available online: http://ftp.eecs.umich.edu/people/jfm/PMCCS-9_Slides.pdf (accessed on 15 November 2021).
5. Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.* 2004, 1, 11–33.
6. Resilient Computing Systems; Anderson, T. (Ed.) Wiley: New York, NY, USA, 1985; ISBN 0471845183.
7. Rosati, J.D.; Touzinsky, K.F.; Lillycrop, W.J. Quantifying coastal system resilience for the US Army Corps of Engineers. *Environ. Syst. Decis.* 2015, 35, 196–208.
8. Ron, R.; Richard, G.; Deborah, B.; Rosalie, M. Draft SP 800-160 Vol. 2, Systems Security Engineering: Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems. 2018. Available online: https://insidecybersecurity.com/sites/insidecybersecurity.com/files/documents/2018/mar/cs03202018_NIST_Systems_Sec (accessed on 16 April 2020).
9. Carias, J.F.; Borges, M.R.S.; Labaka, L.; Arrizabalaga, S.; Hernantes, J. Systematic Approach to Cyber Resilience Operationalization in SMEs. *IEEE Access* 2020, 8, 174200–174221.
10. Hopkins, S.; Kalaimannan, E.; John, C.S. Foundations for Research in Cyber-Physical System Cyber Resilience using State Estimation. In Proceedings of the 2020 SoutheastCon, Raleigh, NC, USA, 28–29 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–2, ISBN 978-1-7281-6861-6.
11. Dyka, Z.; Vogel, E.; Kabin, I.; Aftowicz, M.; Klann, D.; Langendorfer, P. Resilience more than the Sum of Security and Dependability: Cognition is what makes the Difference. In Proceedings of the 2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10–14 June 2019; Stojanovic, R., Ed.; IEEE: Piscataway, NJ, UAS, 2019; pp. 1–3. ISBN 978-1-7281-1739-3.
12. Nygard, M.T. Release It! Design and Deploy Production-Ready Software, 2nd ed. 2018. Available online: <https://www.oreilly.com/library/view/release-it-2nd/9781680504552/> (accessed on 15 November 2021). ISBN 9781680502398.
13. Jean-Claude, L. From Dependability to Resilience; DSN: Anchorage, AK, USA, 2008; p. 8. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.331.8948&rep=rep1&type=pdf> (accessed on 15 November 2021).

14. De Florio, V. On the Constituent Attributes of Software and Organizational Resilience. *Interdiscip. Sci. Rev.* 2013, 38, 122–148.
15. Jen, E. Stable or robust? What's the difference? *Complexity* 2003, 8, 12–18.
16. Clark-Ginsberg, A. What's the Difference between Reliability and Resilience? 2016. Available online:
17. Dyka, Z.; Vogel, E.; Kabin, I.; Klann, D.; Shamilyan, O.; Langendörfer, P. No Resilience without Security. In 2020 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 8–11 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5. Available online: <https://ieeexplore.ieee.org/abstract/document/9134179> (accessed on 14 November 2021).
18. Deborah, B.; Richard, G.; Jeffrey, P.; Rosalie, M. Cyber Resiliency Engineering Framework. 2011. Available online: https://www.mitre.org/sites/default/files/pdf/11_4436.pdf (accessed on 16 April 2020).
19. World Economic Forum. A Framework for Assessing Cyber Resilience; World Economic Forum: Geneva, Switzerland, 2016.
20. Castano, V.; Schagmayev, I.; Schagaev, I. Resilient Computer System Design; Springer: Cham, Switzerland, 2015; ISBN 9783319150680.
21. Hollnagel, E. Prologue: The scope of resilience engineering. In *Resilience Engineering in Practice: A Guidebook*; Hollnagel, E., Pariès, J., Woods, D., Wreathall, J., Eds.; Aldershot: Hampshire, UK, 2011; pp. 29–39.

Retrieved from <https://encyclopedia.pub/entry/history/show/38912>