# Trustworthiness in IoT Systems

## From Design to Operation

2022-07-27

**Author:**

Marcellus Buchheit
WIBU-SYSTEMS
*mabu@wibu.com*

## CONTENTS



## FIGURES



## TABLES

# 1 INTRODUCTION

This article is the sequel to the author's former article *Trustworthiness in Industrial System Design [1]* which addressed trustworthiness in the design of industry IoT systems, introducing the *Trustworthiness Method* as an important trustworthiness implementation technique and the *Trustworthiness System Status* for assigning trustworthiness methods to keep the system in a specific status.

This new article extends the usage of trustworthiness from the design towards the actual operation of industry IoT systems: During the design of the system, most trustworthiness aspects are seen as "static": Expectations on how the future system will behave and how trustworthiness will be considered and addressed. During the operation of the same system, trustworthiness aspects are growingly "dynamic": Unexpected incidents may weaken the trustworthiness of the system and need to be instantly addressed and avoided in the future by enhancing the system.

If the reader is totally unfamiliar with the concept of trustworthiness, *[2]* will be great introduction.

Before looking deeper into the operational details, the Trustworthiness Method from *[1]* will be introduced again in the next chapter.

# 2 TRUSTWORTHINESS METHODS

The first challenge of using trustworthiness in an industry IoT system is that none of the trustworthiness characteristics can be implemented as a separate technology and that the trustworthiness of the system cannot be implemented by just combining such technologies: The characteristics may support or block each other: a simple combining of trustworthiness characteristics does not lead to a real trustful system.

The solution is to take the system away from the trustworthiness characteristics and move to methods which are assigned to the specific parts of the system. In traditional systems, such methods had been used extensively but were not classified by the Trustworthiness Characteristics. And this classification can be extended by other attributes.

*Definition:* A Trustworthiness Method is defined as a component, tool, technology, software application, operational procedure, or a management directive which is assigned to at least one trustworthiness characteristic.

Such methods are referred to as the *Trustworthiness Safety Method*, *Trustworthiness Resilience Method,* etc. If a method is assigned to several trustworthiness characteristics, the list of characteristics is separated with a slash e.g., *Trustworthiness Security/Privacy Method*.

The definition of such a method is intentionally as broad as possible as only the assignment to one or more trustworthiness characteristics is key.

Examples of Trustworthiness Methods are:

- *Fire extinguisher*: a tool and a Trustworthiness Safety Method.
- *$CO_2$ fire suppression system*[1]: a tool and a Trustworthiness Resilience Method (the main purpose is to protect the system rather than the environment or humans; $CO_2$ is indeed dangerous for humans).
- *Network firewall*: tool and a Trustworthiness Security Method.
- *Melt-resistant steel*: technology and a Trustworthiness Resilience Method.
- *Windmill Restart*: operational procedure for airplanes during an engine flameout and a Trustworthiness Resilience Method[2].
- *Electric motor brush replacement*: operational procedure and a Trustworthiness Reliability Method.
- *Brushless motor*: technology and a Trustworthiness Reliability Method.
- *Encryption of all social security numbers on servers*: management directive and a Trustworthiness Privacy Method.

Examples of Trustworthy Methods assigned to several trustworthiness characteristics are:

- *Fire-resistant plastic*: technology and a Trustworthiness Safety/Resilience Method: it prevents a fire from spreading and endangering humans (safety) but also prevents the system itself from damage (resilience).
- *Using encrypted hard disks*: management directive and a Trustworthiness Security/Privacy Method.

Most of these Trustworthiness Methods for Industry IoT systems have existed for many years; the only novelty being the assignment to one or more of the trustworthiness characteristics and the addition of a new name.

# 3 WHAT DOES OPERATION MEAN?

In the world of IoT systems there is typically a differentiation between IT (Information Technology) and OT (Operational Technology) and the merge of both is called IT/OT convergence; the concept has been frequently addressed in several IIC documentations including a reference to trustworthiness *[3]*, see Figure 3-1. This IT/OT convergence brings technology from IT (for example computer networks, databases, or cloud services) into OT, which is addressing a world of physical execution (like building products in an assembly line or mining ore), unknown in the IT world.

[1] Gaseous fire suppression, *https://en.wikipedia.org/wiki/Gaseous_fire_suppression* and Carbon dioxide, https://en.wikipedia.org/wiki/Carbon_dioxide

[2] Flameout, https://en.wikipedia.org/wiki/Flameout]

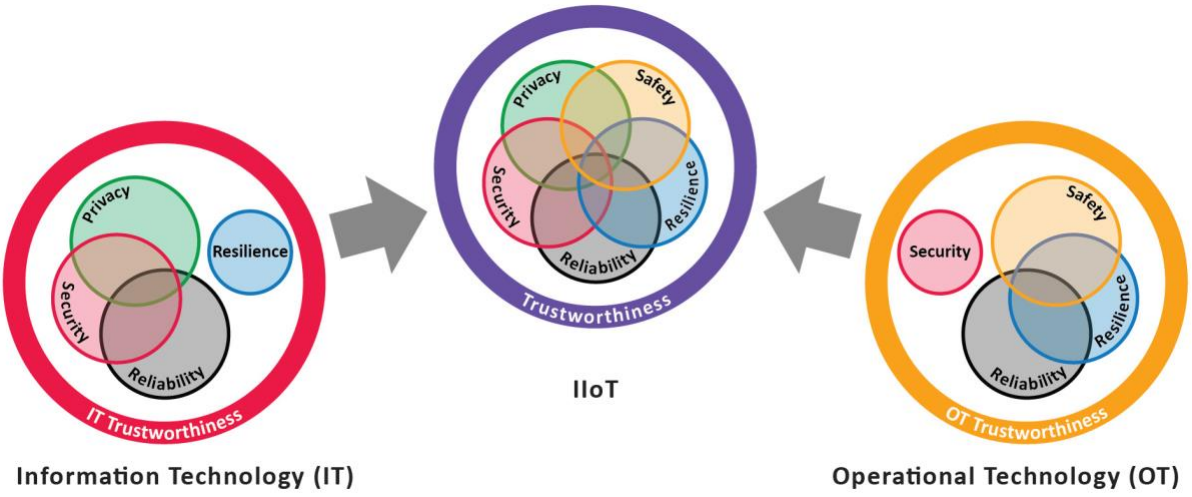


Figure 3-1: IT/OT convergence and trustworthiness.

Even before the days of digitization, traditional OT was well versed on the importance of the concepts of safety, reliability, and resilience of the operation.

In the early days of computing (starting in the 1950s), computer executions were batched: Data input was uploaded as punch cards into the computer and the output came back as a print-out after a reasonable time. There was no direct interaction with humans and no process of automatic execution in loops with conditions. Thus, the concepts of reliability and resilience of the service were not a high priority. Real interaction with users started many years later with terminals and today–with network-connected systems, internet communication, cloud computing, and websites–IT has become fully operational.

An example is *real-time* behavior: Visitors to websites and clients calling cloud services or database requests all expect reasonable response times with a maximum timeout. Cloud services are extremely reliable: For example, the consequences of unexpected hardware crashes or software bugs can be solved with quick automatic restarts inside the service, mostly invisible to the client. Another example is resilience: If a web service, representing an on-line shopping system is partially down, it most likely can still accept orders even though the automatic processing is delayed or temporarily switched to manual execution by humans.

So many demands from traditional OT like real-time behavior, high reliability, or resilience are already well-understood by implementers of modern IT and no longer part of any IT/OT convergence. Therefore, the IT side can deliver best practices around such things to the OT side.

But safety requirements from the OT side are relatively new to the IT side. The reason for safety is based solely on *physical* operation of the OT side, which is still unknown and hardly understood by the IT side, even with a lot of experience in *digital* operation. So, a better understanding of the whole IoT landscape would be distinguishing between traditional (non-operational) IT with file management, emails, web browsing, etc. and Operational IT with real time and reliability

demands. On the OT side we have some area of strictly digital implementation, called Digital OT; and finally, Physical OT with the risk of physical damage, injury and death, shown as IT/OT Landscape in Figure 3-2. Table 3-1 shows significant differences between the three areas of industry IoT. Many implementation details, opportunities and challenges in Operational IT and Digital OT are identical or very similar, helping to reduce the gap between OT and IT.

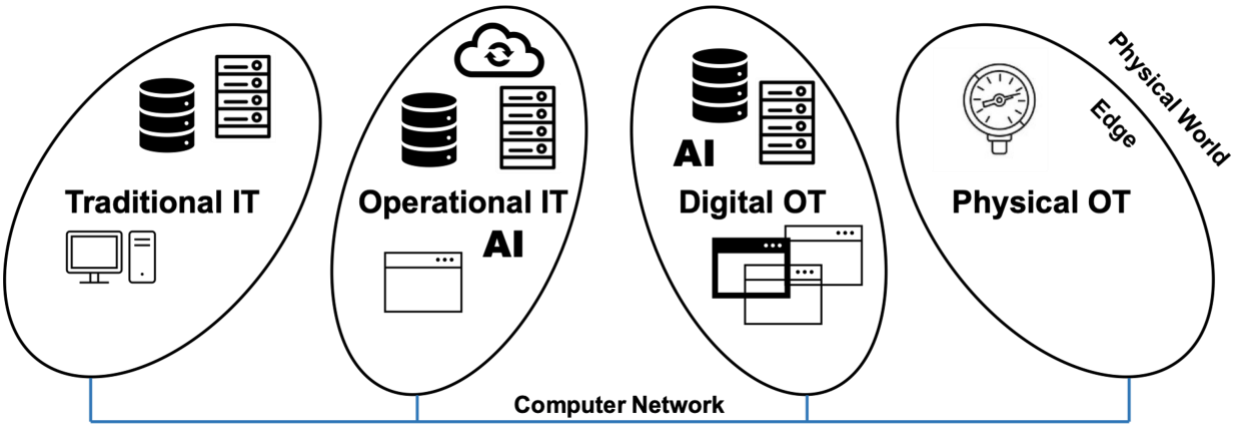


Figure 3-2: IT/OT landscape.

| Context | Traditional IT | Operational IT | Digital OT | Physical OT |
|---|---|---|---|---|
| Real-time behavior | human-based | system-based | system-based | physical-based |
| Data complexity | high | high | high | low |
| Reliability | medium | high | high | very high |
| Safety | none | none | medium | very high |
| Security | very high | very high | very high | high |
| Resilience | very low | medium | medium | high |
| Privacy | very high | very high | high | medium |

Table 3-1: Comparison of traditional IT, operational IT, digital OT and physical OT.

Merging digital and physical operations is a new type of convergence, which is better understood within the context of trustworthiness, see Figure 3-3.

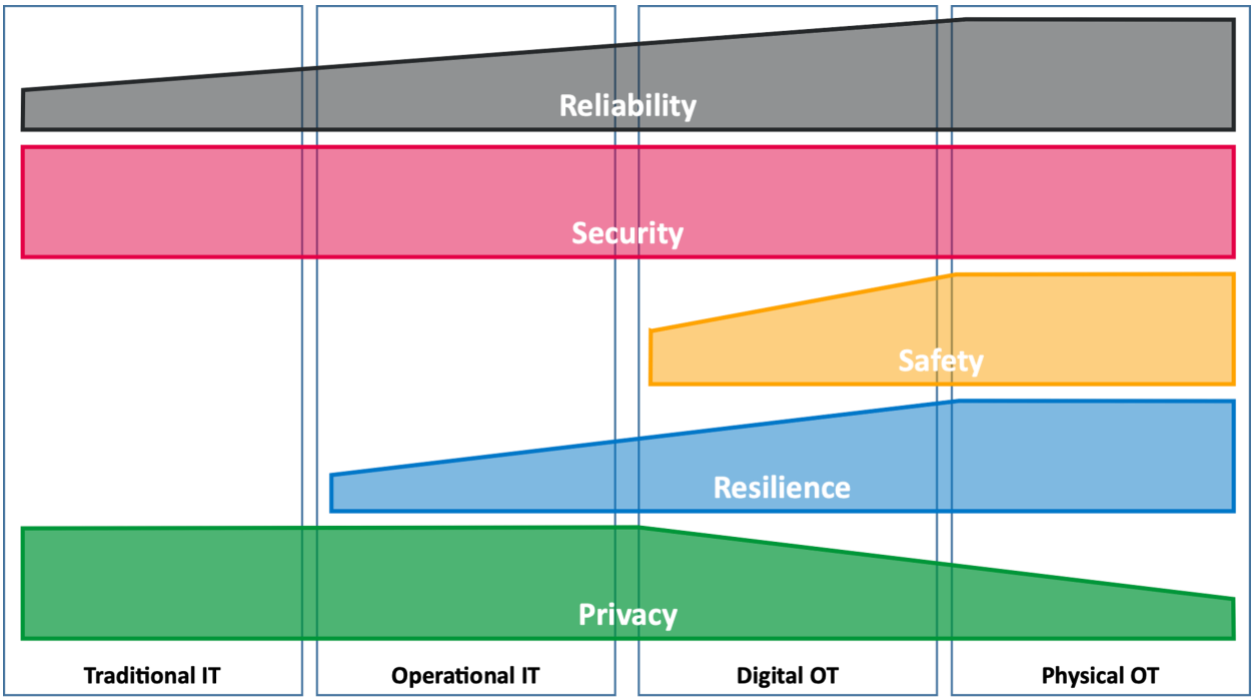


Figure 3-3: IT/OT landscape and trustworthiness.

# 4 TIME. HAZARDS, PERILS

The former article *[1]* introduced the *Trustworthy System Status Model* (TSSM), Figure 4-1 shows this model with the original diagram. All incidents which likely challenge the *normal* status of the system are driven by threats. The term *threat* is widely used in security models, borrowed from the military context with attacks and defense.

However, in the world of safety, reliability and commercial resilience models, the usage of *threat* is rare. Instead, the term *hazard* is more common. With the merging of security and safety in trustworthiness, both terms will be used side by side. In the normal usage of the English language, both terms are synonyms, typically describing the potential start of an accident or an attack, leading to equipment damage, injury, or death of humans. Definitions in dictionaries like Collins *[2]* or Merriam-Webster *[3]* are less about the reason of a threat or hazard but more about the results: source of danger, causing damage, etc. – and used synonymously for hazard and threat.

In this article both terms will be defined in a unique way to be strictly used to describe any incident in a system caused either by a threat or by a hazard. In the same way, terms like attack or accident are properly defined and assigned.

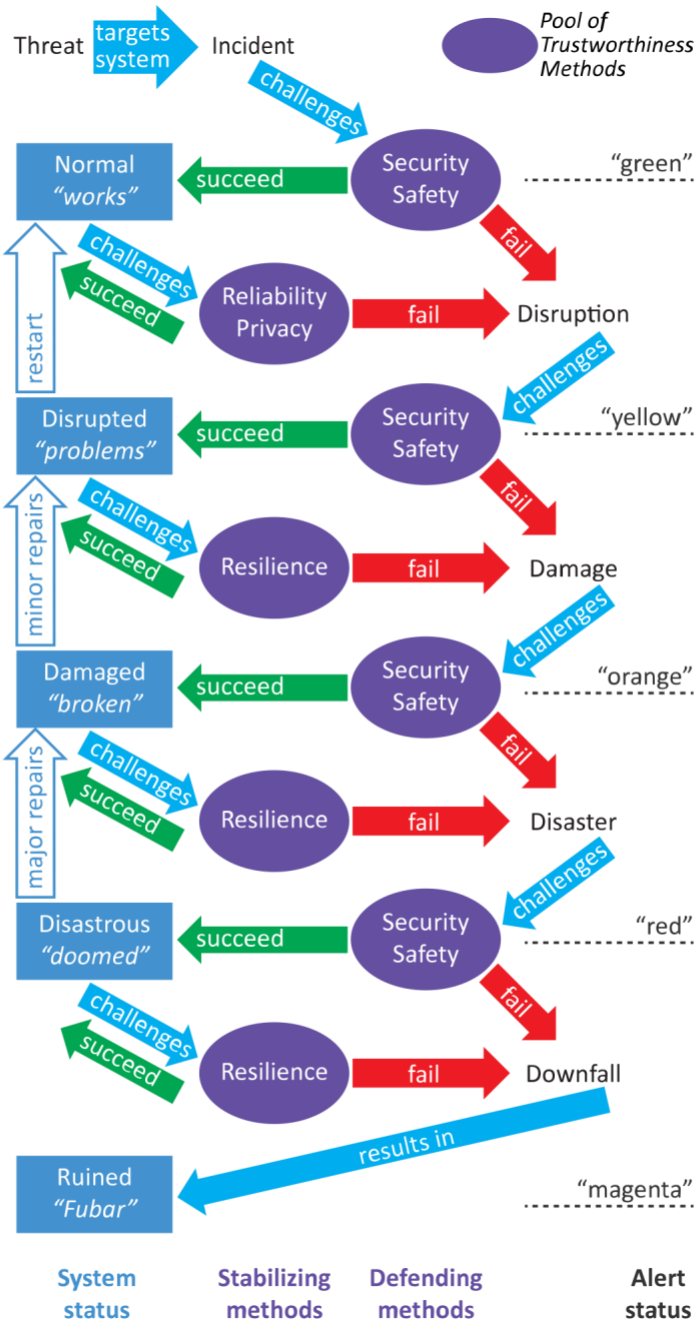


Figure 4-1: Trustworthy system status model (TSSM).

## 4.1 HAZARD

The dictionary definitions of hazard are very vague about the cause of a hazard. For example, Collins *[2]* defines *hazard* as:

- a hazard is something which could be dangerous to you, your health or safety, or your plans or reputation. *("Countable Noun")*

In principle in this definition the word *hazard* could be replaced by *threat* and the definition would still be correct. That's why in the context of trustworthiness of a system both definitions should be distinguished by assigning the following attributes to hazards:

- a hazard may lead to an incident in an unintentional, random way.
- a hazard may be well-known or unknown and hidden or visible.

In general, a system is protected against hazards with Trustworthiness Methods: If the process inside the system requires protection (preventing a disruption), they are Reliability Methods; if humans need to be protected from harm of a hazard, they are Safety Methods; If personal information needs protection, they are Privacy Methods; and if the system itself requires protection, they are Resilience Methods.

If such methods cannot defend successfully against a hazard-caused incident, the status of a normally running system leads to disruption. And if the hazard cannot be stopped in the status of the interrupted system, there is a risk of damage or even total loss of the system. A good example could be an overheated battery in a robot which leads to an unexpected fire in the production which finally burns down the whole production facility. All such incidents, caused by hazards, are defined as *accidents*.

## 4.2 THREAT

The dictionary definition of the term *threat* presents three different meanings. Collins *[2]* defines it as:

- A threat to a person or thing is a danger that something bad might happen to them.
- A threat is also the cause of this danger. *("Variable Noun")*
- A threat is a statement by someone that they will hurt you in some way, especially if you do not do what they want. *("Countable Noun")*

The first definition is very general: Any incident which challenges a system is also a threat for this system. Any hazard could be such a threat as well. That is why this definition is not useful for our purpose. The second and the third definitions are more appropriate:

- A threat is a cause of danger - in our case an incident.
- A threat is coming from a "someone." This most likely refers to a person (or a group of people, like an army in a war) but could also be an autonomous robot (for example, a self-controlling drone or a military or criminal cloud service which automatically attacks a system as soon as a security weakness is discovered).
- A threat is a *statement*, which means it is intentional and not random.

In general, a system is protected against hazards with Trustworthiness Security Methods. If such a method cannot block an attack-related incident, the status of a normally running system leads to disruption. Similarly, if the hazard cannot be blocked, the result may be more critical status

changes like damage or disaster as shown in Figure 4-1. Such incidents are then defined as *attacks*.

## 4.3 INCIDENT

Neither a hazard or threat leads imminently to an accident or attack, but both have the potential for creating a real problem. For example, nearly any physical system has a fire-hazard, or a loss-of-power-hazard and any internet-connected system has the threat of a hacker-attack. That moment when a system is really affected by a hazard or threat is considered an *incident*.

## 4.4 OBVIOUS DIFFERENCES: HAZARD VERSUS THREAT

We see clear differences between hazards and threats in the way there are introduced in the text above:

- A hazard may lead to an accident but never to an attack.
- A threat may lead to an attack but never to an accident.
- A hazard-caused incident is random and not intentional.
- A threat-caused incident is intentional and not random.
- Trustworthiness Reliability/Safety/Privacy/Resilience Methods protect the system from hazard-caused incidents.
- Trustworthiness Security Methods protect the system from threat-caused incidents.

For example, a Trustworthiness Privacy Method could demand as an operational directive that all files containing social security numbers, are permanently encrypted, except when they are viewed or edited e.g., Excel files in Excel. And another operational directive specifies that social security numbers cannot be sent via external email. However, an incident could occur where an employee erroneously forgets these directives and copies social security numbers out of Excel via clipboard into an external email.

To prevent this incident from causing a severe accident of violating privacy by emailing, a mail server extension service could scan all outgoing emails for information looking like social security numbers and interrupt the sending operation. Such a blocker would also prevent any employee from trying to intentionally send out the social security numbers as part of a hacker attack, likely warning security departments about the incident as well. This blocker is an additional Trustworthiness Privacy/Security Method but realized as a software tool, not just an operational directive, and preventing accidents and attacks.

A summary of the differences between hazards and threats is shown in Table 4-1.

| Context | Hazard | Threat |
|---|---|---|
| Result of an incident | Accident | Attack |
| Cause of incident | Random or direct consequence of another accident | Intentional or direct consequence of another attack |
| Protecting Trustworthiness Methods | Reliability, Safety, Resilience or Privacy | Security |

Table 4-1: Terms and definitions of the peril model.

### 4.5 Software Bug

All IoT systems use software to control the flow of data. The reason that a software module does not work as expected is known as a *Software Bug.* In general, Software Bugs are design or implementation flaws (in many cases a result of poor programming practices) but assumed to be unintentional - no serious software designer or code will implement software bugs intentionally. And results of such bugs – crashing of code-executing software modules – are defined as accidents and not as attacks. That's why in the context of trustworthiness, software bugs are hazards and not threats.

### 4.6 Nature-Caused Incidents

Another gray area is nature-caused incidents, e.g., when a physical system is hit by a heavy windstorm, unusually hot weather, or an earthquake. Are such incidents caused by hazards or threats? Human or autonomous robots do not intentionally start such incidents, and because they may lead to accidents, they generally are not seen as attacks to the system. That's why in the context of trustworthiness nature-caused incidents are caused by hazards and not by threats.

### 4.7 Peril

The dictionaries generally define *peril* as "great danger" (Collins) or "exposure to the risk of being injured, destroyed, or lost" (Merriam-Webster) without defining any specific reason. This makes this word suitable to be used in the context of trustworthiness as an umbrella term for hazard and threats:

Any system incident is caused by one or more of its perils, which can either be a threat or a hazard.

## 5 A System Perils Model

So far, we have introduced the terms *incident*, *hazard*, *accident, software bug*, *threat*, *attack,* and *peril*. They all appear in visual relation in Figure 5-1. Table 5-1 shows the definitions of these terms proposed by this article.

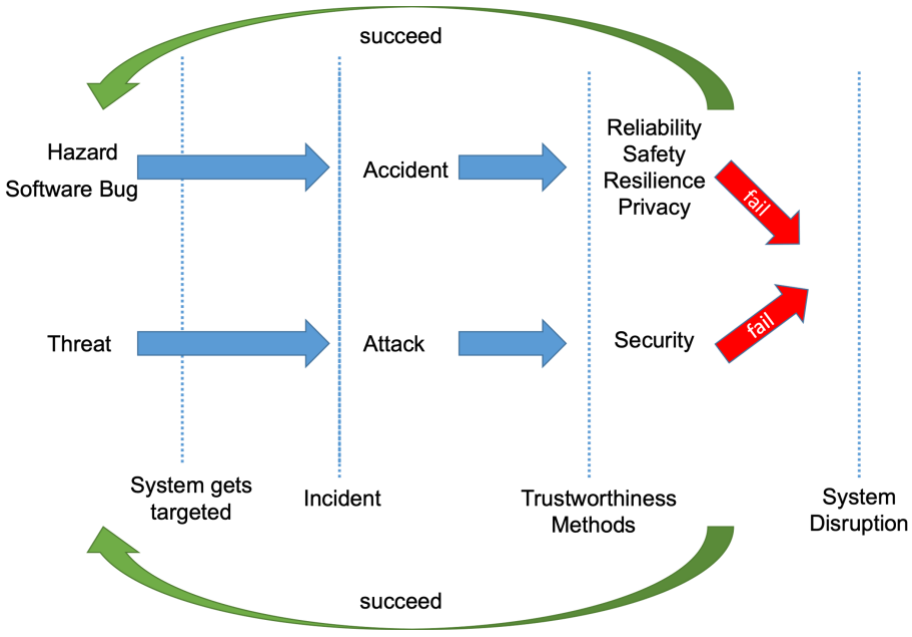


Figure 5-1: System perils model.

| Term | Definition |
|---|---|
| Incident | The event that a peril targets the system. |
| Hazard | A peril which results in an accident if it targets the system. A hazard occurs randomly and may be visible or hidden. |
| Software Bug | A hazard in the design or implementation of software. |
| Threat | A peril which results in an attack if it targets the system. A threat occurs intentionally and is mostly visible but may be hidden in rare cases. |
| Peril | A peril is either a hazard or a threat. All specific hazards and threats to a system are the *Perils of the System* |
| Accident | The result of a hazard-caused incident. The system should be protected with a Trustworthiness Reliability, Safety, Resilience, or Privacy Method. |
| Attack | The result of a threat-caused incident. The system should be protected with a Trustworthiness Security Method. |

Table 5-1: Terms and definitions of the peril model.

# 6 A REFINEMENT OF THE TSSM

The new graphic distinguishes between hazards and hazards with the corresponding Trustworthiness Methods, see Figure 6-1.

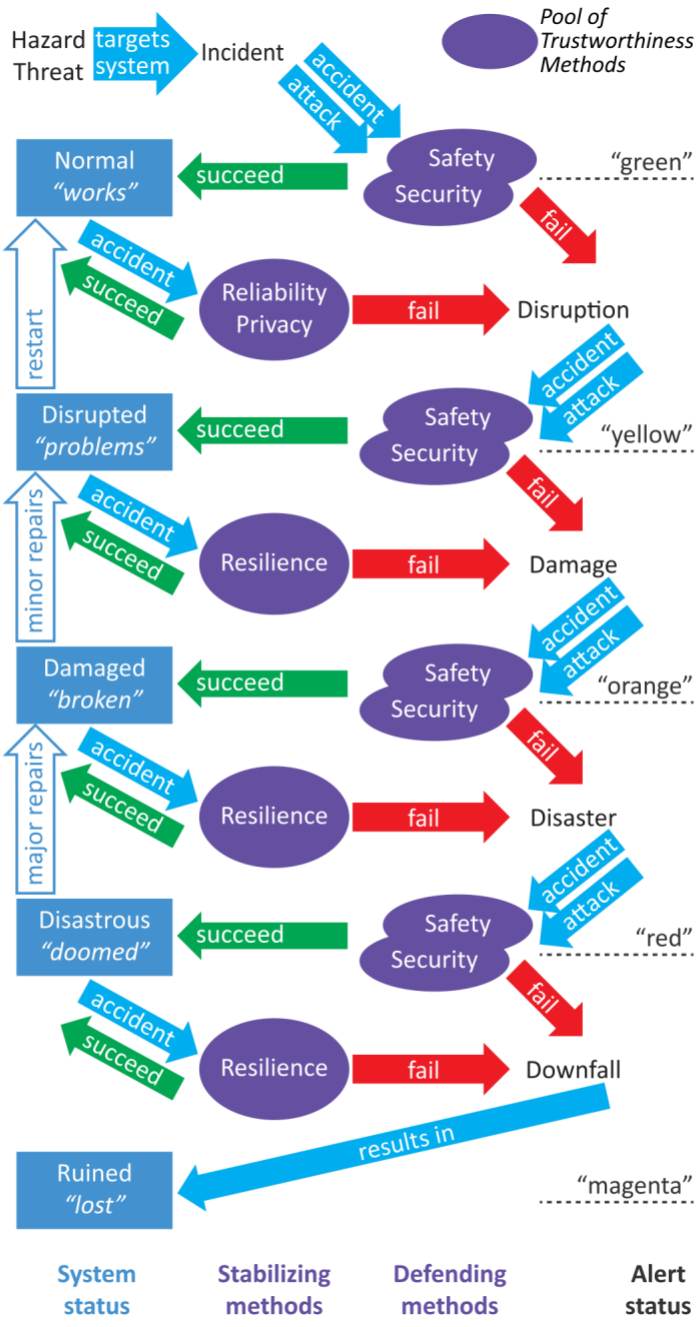


Figure 6-1: Refined trustworthy system status model (TSSM) with hazards and threats.

The major revision to the graphic in Figure 6-1 is the assignment of Trustworthiness Security Methods to Threats and the other four trustworthiness method types to hazards.

## 7 System Status Beyond the TSSM

The TSSM shows only five different status levels. In practice, any system designer or operator can add *minor* levels between the *major* levels, which creates specific *intermediate* levels. For example, not every non-addressed peril in a disrupted system leads automatically to a damaged

system. A Trustworthiness Resilience Method most likely allows the execution of the system with limited capabilities even after the disrupted status is reached. But these “minor” levels follow the same graphical schema as shown in the TSSM.

It is important to understand that an initial threat which leads to an attack can also lead instantly to an additional accident. In this specific case, the Trustworthiness Security Method, which should protect the system from this threat, has a hazard (which was probably hidden but suddenly visible after the incident caused by this threat happened the first time). And due to this accident, a new threat could cause an even bigger attack. Which means, at the time of an incident, a cascade of hazards in the system could quickly convert a harmless attack into a more potent and dangerous attack.

# 8 PREPARING FOR THE INCIDENTS

After a system is designed and setup, the operator takes over the whole responsibility to run the system in a trustworthy way. During the design, all visible (well-known) perils should be specified and addressed by specific Trustworthiness Methods, and everything should be documented for the operator. The following details must be specified:

- Type of peril (hazard or threat)
- Source of peril (specific component, external location etc.)
- Likelihood of impact as incident
- Detailed description of the peril
- Prepared Trustworthiness Methods to reject this peril
- Describing the specific functionality of the Trustworthiness Methods for these peril
- Consequences if this Trustworthiness Method will fail: This will lead to other (referenced) perils, in many cases more severe.

Furthermore, the operator must practice what happens if such perils target the system as incidents. However, many of these incidents will lead to a disrupted system, so practicing the incident operations can only be done outside of the normal usage of the system, which is not always practical, especially not in systems running 24x7. In this case, a simulation of the system, based on its Digital Twin, would help tremendously.

During such incident practices, new perils will most likely appear, because during the design phase, the impact of a specific peril was not completely visualized. In this case, the operational management must refine and expand the documentation around incidents.

Another reason to expand the list of perils is the detection of hidden hazards. They usually arise during a real incident. In such cases, the down time of a disrupted system is longer than usual because the selection of the correct trustworthiness method to stop this hazard is frequently not instantly apparent and sometimes a trial-and-error approach may be required. But after the

hazard has been stopped and the system status is back to normal, the correct operations to prevent this hazard in the future must be documented as well.

Most threats to a system are well-known and should be already documented during the design phase. But in the world of cyber-attacks, in particular, new attack methods often appear, and new crime variants become popular. An example is ransomware. It was hardly used before 2005 [Wikipedia Ref] and has since expanded into different variants of crime: Originally after paying the ransom, the blocked information was "returned" by supplying a decryption key. Today, there is also the threat that the locked business information could be sold to competitors or just delivered to the public if the ransom is not paid. Again, such new threats need also to be identified, documented, and addressed with effective Trustworthiness Methods.

## 9 SUMMARY

The expansion of the well-known IT/OT model to four areas – Traditional IT, Operational IT, Digital OT, and Physical OT will help reduce the gap in thinking and implementing between the IT and OT world of an Industry IoT system. Trustworthiness Methods can be better assigned to these four areas rather than just to IT and OT; many of them will either overlap Traditional IT and Operational IT, Operational IT and Digital OT, and finally Digital and Physical OT.

This article also introduces the System Peril Model: In the past, only threats were seen as challenges to a trustworthy system. But now we strictly separate between threats and hazards and the results with attacks and accidents. Moreover, trustworthiness characteristics are clearly assigned to these perils: Security to threats and attacks, Safety, Reliability, Resilience and Privacy to hazards and accidents.

## 10 REFERENCES

[1] Marcellus Buchheit: Trustworthiness in Industrial System Design, Industry IoT Consortium, Journal of Innovation, Fall 2018

[2] Marcellus Buchheit, Frederick Hirsch, Sven Schrecker: A Short Introduction into Trustworthiness, Fall 2018

[3] Industry IoT Consortium: Industrial Internet of Things Volume G4: Security Framework, version 1.0, 2016-September-26, *https://www.iiconsortium.org/IISF.htm*

## 11 ACKNOWLEDGEMENTS

The views expressed in the *IIC Journal of Innovation* are the contributing authors' views and do not necessarily represent the views of their respective employers nor those of the Industry IoT Consortium®.

➢ Return to *IIC Journal of Innovation landing page* for more articles and past editions.