How IT Infrastructures Break: Better Modeling for Better Risk Management

Benjamin Somers, Fabien Dagnat, Jean-Christophe Bach

How IT Infrastructures Break: Better Modeling for Better Risk Management

Benjamin Somers[1,2][0000−0002−0359−0902], Fabien Dagnat[1][0000−0002−2419−7587], and Jean-Christophe Bach[1][0000−0001−6986−1093]

[1] IMT Atlantique, Lab-STICC, UMR 6285, F-29238 Brest, France
[2] Crédit Mutuel Arkéa, 29480 Le Relecq-Kerhuon, France

Abstract. IT infrastructures break. Whether it be computer attacks or software, human or hardware failures, IT safety and security risk is present in many technical and organizational domains. Risk management is therefore essential to ensure infrastructure resilience, compliance with legal and contractual requirements and a better knowledge of what causes what. But risk management is hard to automate, sometimes because criteria are subject to human appreciation, sometimes because of an incomplete or wrong knowledge of the infrastructure itself. And this latter factor has become more evident with the advent of modern cloud-native architectures: complex and dynamic infrastructures make risk assessment difficult. In this article, we propose an approach based on infrastructure modeling to help automate the risk assessment process for IT infrastructures. Instead of focusing first on hazard analysis, our approach attempts to consider (most of) such an analysis as a consequence of infrastructure modeling. By deciding to focus on the infrastructure modeling itself and by involving as many of the company's stakeholders as possible in the process, we intend to make risk assessment more collaborative and thorough, by taking advantage of everyone's expertise.

Keywords: Distributed Infrastructures · Infrastructure Modeling · Risk Assessment

1 Introduction

Over the past few decades, the presence of IT in critical infrastructures has been continually on the rise [22]. This has allowed for unprecedented growth in various business sectors, but has brought its share of risk into the equation. From a few computers with monolithic software, IT infrastructures have evolved towards data centers with several thousands of servers running interacting microservices. These interactions themselves are becoming increasingly complex [24] as the design of such infrastructures follows a tendency towards abstraction. From bare metal to virtual machines, from physical networks to virtualized fabrics, from local storage to distributed datastores, risk is more and more difficult to assess [19] as there is a clear break between physical and virtual infrastructures.

On top of these highly dynamic virtual infrastructures lie reconfigurable software packages. Components not known before runtime (such as dynamic

libraries) are used to build software which now rely heavily on external services. As IT gets more and more ubiquitous, we observe a multiplication of entry points, inducing a much larger attack surface [12]. Assessing the risk requires to combine together information collected from a large number of stakeholders, not always willing to share.

Risk management is not something to be taken lightly: sometimes, lives depend on the very systems being studied [34]. So methods such as FTA and FMEA [28], HAZOP [14] or STPA [18] have been devised in that regard. In parallel, the various industries have been developing safety and security regulations to guide such risk analyses, whether in aerospace [29] or IT [10]. Finally, as IT services have multiplied, *Service-Level Agreements* (SLAs) have emerged and various certifications have been introduced to increase customer confidence and guarantee a certain quality of service [13].

But to be able to certify such infrastructures, one has to understand their components (technical, such as computers, and also human) and their interactions. It can be done with the help of modeling frameworks from the enterprise modeling communities (such as RM-ODP [1] or Archimate [31]) to the formal methods communities (such as Alloy [17] or Petri nets [23]) to software and network engineering communities (such as UML [25] or OMNeT++ [32]).

In this article, we advocate a collaborative approach to risk management for IT infrastructures based on better modeling of such infrastructures. To this end, we present the theoretical background of our study in section 2. Section 3 proposes a set of methods to support risk management in distributed infrastructures. In section 4, we show an example with the advantages and limitations of our technique. Finally, the article finishes with a conclusion in section 5.

2 Related work

In this section, we present the scientific context of our study, in the fields of risk management and infrastructure modeling. Our work attempts to unite the different methods presented here without changing the approaches and tools already in place: a federation of models respecting the expertise of each actor.

2.1 Risk analysis

The ISO 31000 standard [16] defines risk management as a five-step process. First, the context of the study has to be defined: *what* the system is, *when* it is considered, *where* it is, *who* is involved, *why* and *how* the study is being done. The next three steps are what the standard refers to as the risk assessment step. It consists first of all of identifying risks and their causes and impacts. Then, a qualitative and quantitative risk analysis is carried out to assess likelihoods, confidence levels and magnitude of consequences. The last step of risk assessment is its evaluation, to compare the results and make decisions. Finally, the risk treatment phase addresses risks by modifying risk sources, likelihoods, consequences or simply

abandoning the activity that led to it. In this work, we focus on the first two steps of the process (context and risk identification).

Several implementations of risk management have been developed, such as *Fault Tree Analysis* (FTA), *Failure Mode and Effects Analysis* (FMEA), *Hazard and Operability study* (HAZOP) or *System-Theoretic Process Analysis* (STPA). There are also risk management methods aimed at information security [2] which implement the ISO 27005 standard [15], more focused on IT risk, such as *EBIOS Risk Manager* (EBIOS-RM).

FTA [28] is a top-down approach: top events are first identified and their causes are recursively analyzed and combined using boolean logic gates. Individual causes are put in OR gates and collective causes are put in AND gates (with care taken to verify the independence of the causes). It is a deductive approach carried out by repeatedly asking: how can this event happen and what are its causes?

In contrast, FMEA [28] lists only single failures, even though they have no high-level impact. FMEA is a bottom-up, inductive method used to identify low-level failures and study their impacts on the higher levels. FTA and FMEA are often used together thanks to their opposite views, but are very time-consuming to apply thoroughly, therefore the analyses are often incomplete [8].

HAZOP [14] works quite differently. It uses standardized questions and words to investigate on possible deviations from a design intent. The technique is a qualitative way to assess complex processes and focuses on structured discussions. In some cases, this format may however provide a false sense of security by being too guided [6].

These bottom-up methods use a divide and conquer approach: they analyze separately each part to provide a synthetic assessment for the system. But interactions between such parts and dynamic behaviors may be overlooked [30].

STPA [18] is a systems approach to hazard analysis. Instead of focusing on failures, it focuses on the losses of control being the real cause of accidents. It is a top-down method to study functional control instead of physical components.

EBIOS-RM [3] proposes a cyclic approach for risk management. The approach consists of five workshops identifying high level, textual risk scenarios and their resolutions with security measures.

Some methods are more suitable for certain domains and we believe that it is important to let different actors in the IT infrastructure choose which ones they want to exploit. As mentioned in [4], a collaborative approach is key, and we believe that federating these approaches is beneficial to the domain.

2.2 Infrastructure modeling

The ISO 31000 standard emphasizes that risk management should be a collaborative endeavor including several domains of expertise to properly define and evaluate risks. This is where we think that infrastructure modeling can be used.

Many languages and representations exist to describe IT infrastructures, from hardware to software, including networks and processes. A datacenter can be described with rack diagrams, illustrating the layout of servers and network components. A piece of software can be represented as UML diagrams [25], to

show its structure and the different interactions at work, or can be described by its code. A network topology can be seen as a mathematical object (such as a graph), described with switch configurations, or as code with *Software-Defined Networking* [20].

However, these languages are mostly static description languages and do not always adapt to the changing structures of modern IT infrastructures. Such languages are particularly adapted to their respective domains [9], but they are sometimes not very understandable by external parties. From the perspective of risk analysis, we believe that we can bring together different technical fields through model federation [11].

3 Guided risk management for IT infrastructures

Over the years, IT infrastructures have evolved from the "traditional" model. Even though so-called legacy systems are still in use in some critical businesses, virtualization has brought a completely different paradigm. Traditionally, software components are directly deployed on physical nodes. This guarantees stability in space (services do not "move" to another place), in quantity (no frequent increase or decrease in the number of nodes), in resources (hardware is not frequently replaced) and network (the topology does not significantly change) over time. A simplified deployment, where functional components are directly mapped to physical servers, is shown on figure 1. Risk analysis here is not much different from what is done in other sectors (although the domain is more prone to component reuse), but it is important to update it as the infrastructure evolves.

Virtualized infrastructures, and more recently cloud infrastructures keep these guarantees of stability for a much shorter period of time (sometimes hours or even minutes). Although instantaneous risk studies are possible, guaranteeing a zero deviation between the studied infrastructure and the real one at any time seems unrealistic. The risk analysis may take longer than the period during which the infrastructure is stable and some infrastructure reconfiguration algorithms are unpredictable or non-deterministic. Moreover, the deployment on these infrastructures (shown on figure 2) is more complex than on traditional ones due to the additional abstraction layer. There is indeed a clear separation between the virtual infrastructures (how VMs "see" one another) and the physical ones (where VMs are located). While the deployment of functions on hardware was *one-to-one* (one functional component per machine) or *many-to-one* (several functional components on the same machine), it is now common to have *one-to-many* (distributed functional component) and *many-to-many* (distributed functional assemblies) deployments. Risk analysis also becomes more complex, as there are now three models (logical, deployment and physical) on which the analysis can be done.

In the rest of this section, we describe what we refer to as the *risk cycle* and present three methods of risk analysis in the context of IT infrastructures.

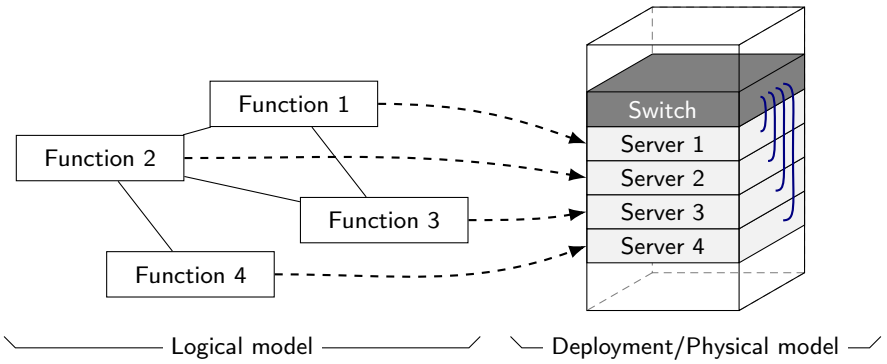


Fig. 1: Traditional IT infrastructure, where the deployment model generally corresponds to the physical model

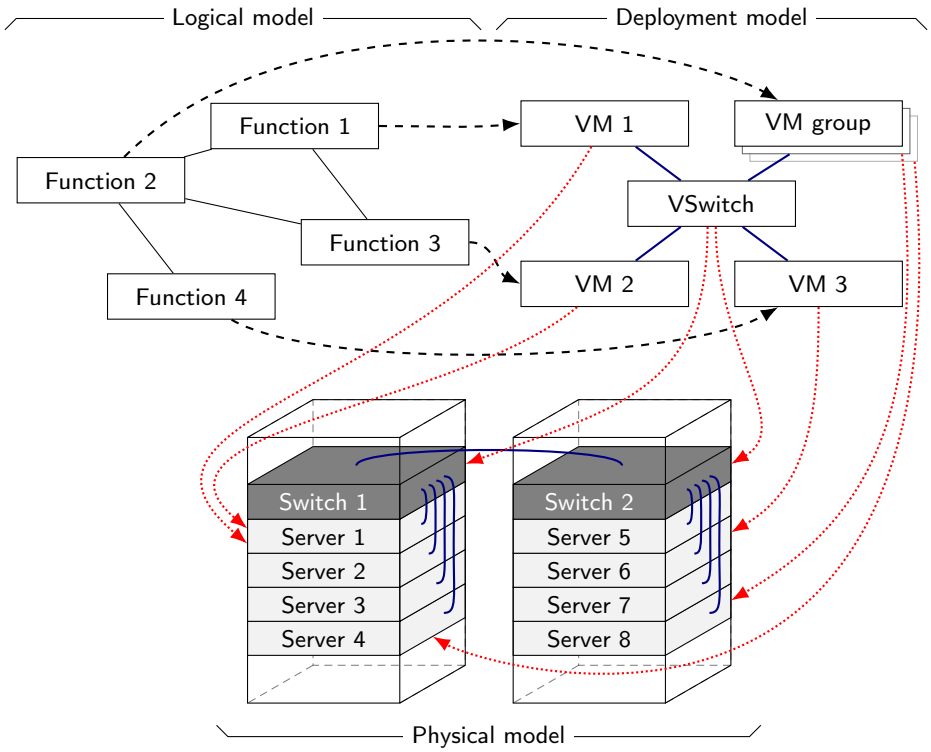


Fig. 2: Modern cloud infrastructure, where there is a clear separation between the virtual and the physical infrastructures

3.1 Side-effect analysis

IT infrastructures are subject to risks. Regulatory bodies, whether they have statutory or advisory powers, provide requirements to follow for adequate risk coverage. These requirements are then implemented or translated into constraints by the company and these constraints can have an effect, beneficial or adverse, on the risk. This is what we call the *risk cycle* (represented on figure 3).

For example, to address the risk of credit card fraud, the Payment Card Industry Security Standards Council proposed the *Payment Card Industry Data Security Standard* (PCI DSS). Depending on the company, these requirements translate into various constraints on the IT infrastructure. For example, PCI DSS sub-requirement 9 (restricting physical access to cardholder data) may be implemented by encrypting the staff's hard drives and deploying a corporate policy prohibiting removable media. These constraints have a direct effect on the risk. An example of an adverse effect is that if employees cannot use USB sticks, they may be more inclined to use insecure sharing services, or even try to disable their corporate antivirus, thus increasing the initial risk.

As another example, to address the availability risk for a single-node database, a team can decide to distribute it over several nodes, which can in turn lead to a consistency risk.

The analysis then continues recursively until the risks are sufficiently addressed or cannot be handled any further. In this case, either they are deemed acceptable (for example, the system is theoretically vulnerable, but too hard to attack in practice) and the analysis can end, or their causes need to be mitigated. We believe this approach to be original, as it focuses on the side effects of constraints instead of parts, processes or systems. Additionally, its stop condition is not the completeness of the analysis *per se*, but rather the diminution of the risk below a certain acceptability threshold defined by the company.

We believe that our approach makes steps that are sometimes implicit in other methods (such as residual risk analysis in EBIOS-RM) more explicit. It also adds a safety dimension in a field where the main focus is on security. Furthermore, the separation of safety/security measures into what has to be done (requirements) and what is actually done (constraints) allows for a better separation of responsibilities.

3.2 Part analysis

Unlike the fundamental domains on which risk analysis was first developed, IT is culturally inclined towards reusability. For hardware, most systems deal with some form of computing power, fast and slow memory and network. Most of the components are also readily available, off-the-shelf [27]. For software, the growing acceptance of open source over the years has led to the creation of many software products that reuse open source libraries, with their fair share of risk [5].

IT infrastructures can thus be made of thousands of components provided by hundreds of actors (companies, communities or single developers). Analyzing all these individual parts is impossible in a reasonable amount of time, as software

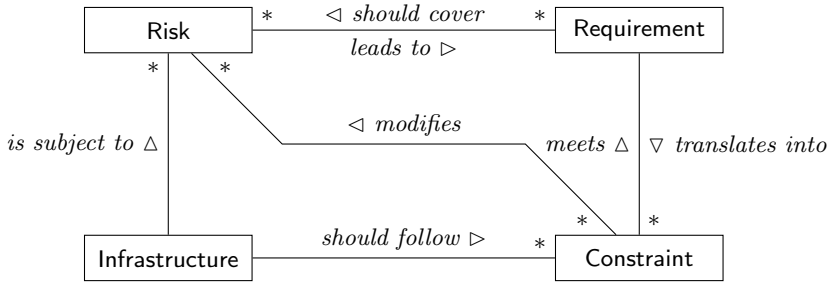


Fig. 3: The risk cycle

and hardware are sometimes seen as a black box (*e.g.* Hardware Security Modules and proprietary firmware) and are regularly replaced or updated.

In such infrastructures, it could prove useful to include the employees in the process of risk assessment. For example,

- Software developers understand the functional interactions between different elements of the software infrastructure and are able to describe precisely with their tools such an architecture;
- Datacenter "smart hands" are hardware experts who know what the servers are made of, in which bay of which datacenter specific machines are, and they have tools for these specific needs;
- Network engineers understand the network topology of the infrastructure, they configure business-critical hardware and, again, have tools and representations to assist their work.

All these actors perform, in their way, infrastructure modeling. Benefiting from such expertise and tools would be a way to involve more people and cover more domains in the risk assessment process. But even with that, the knowledge of the infrastructure may still not be exhaustive and a compromise needs to be found: the risk should be analyzed as best as possible, despite the inaccuracies and lack of knowledge inside the company.

However, knowledge does not need to come only from the company itself. Manufacturers and external developers have indirectly great responsibilities within the company and we believe that a common interchange format for risk analyses could be of great benefit. It could indeed lead towards a more systematic risk analysis that respects the expertise (the actors should "know best") and the responsibilities (if a system breaks, it would be easier to identify who is at fault) of each stakeholder. The individual components of the infrastructure could then be federated in an FMEA-like analysis, particularly well adapted to study the parts of a system. If a software developer or a hardware manufacturer discovers a vulnerability, they can easily update their risk analysis, which in turn updates the risk analysis of a company using these products, without any intervention from them. This approach is a work in progress.

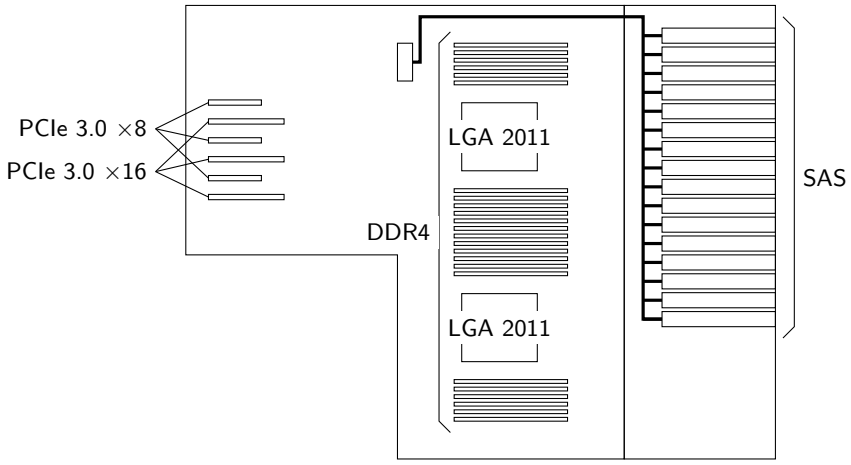


Fig. 4: Typical server motherboard, with memory (DDR4) and expansion card (PCIe 3.0) slots, CPU sockets (LGA 2011) and drive trays (SAS)

3.3 Assembly analysis

Composite hardware, such as motherboards, often have clearly distinct slots for distinct components. In other words, given the risk of a motherboard malfunction, there is a requirement for hardware compatibility, and one of the constraints chosen to implement this requirement is the presence of different sockets. This is illustrated on figure 4.

However, this sole constraint is often not sufficient to eliminate the risk. Old PCIe expansion cards may not function properly on modern hardware, because of deprecation or standard incompatibilities. In addition, the very way of assembling the components on the motherboard has an impact. For example, the layout of the RAM modules can be very constrained on some motherboard models. That is why manufacturers release technical guides on how to build the systems.

Similarly, a piece of software may require a connection to a database, and the communication protocol will identify a type of database. However, if the database engine version is not adequate, the software may fail. This is why software vendors provide installation and deployment guides.

As shown in the two previous paragraphs, components assembled together sometimes exhibit properties that none of the individual components have, this is why both bottom-up and top-down approaches to risk analysis are used. Here again, we could benefit from a common risk interchange format, because actors could perform analyses on the hardware they manufacture, using for example FTA or STPA. If a large-scale infrastructure is suitably modeled, these analyses could be reused in an automated analysis of the whole infrastructure.

4 Case study: a cloud infrastructure

We illustrate in this section how the combination of the above techniques help assess the risk. As risk analysis is an iterative process, our examples are not meant to be comprehensive.

Let us apply these three methods on a fictional European banking company, processing financial transactions and handling client information. The company opted for a private cloud-based architecture for all of its new projects. This cloud infrastructure is linked to the company's legacy systems and communicates with interbank and payment networks. But this is only a part of the its network flows: the company manages a fleet of several thousands of *Point of Sale* (POS) terminals and *Automatic Teller Machines* (ATM) that need a safe and secure connection. On top of that, it provides *Business-to-Business* (B2B) services to other companies wishing to offer banking solutions to their clients.

4.1 Requirements

Because of its activities, the company is subject to various requirements. We have classified some of them below[3] into two categories and four subcategories:

- External
  - Regulatory and legal, with:
    - (R1) *General Data Protection Regulation* (GDPR) [10], governing the use and collection of personal information,
    - (R2) PCI DSS [26], stating how cardholder data can be used and stored;
  - Contractual, with
    - (R3) payment processor requirements [21,33], applying financial penalties in case of non-compliance with quality standards,
    - (R4) SLAs with its business clients, enforcing adherence to various availability and quality criteria;
- Internal
  - Functional:
    - (R5) the services have to "work reasonably well";
  - Technical:
    - (R6) the services have to "be properly secured".

Each of these requirements can be expressed with varying levels of detail. On one side, there are very specific requirements, such as (R3), with clearly defined quantitative criteria. On another side, there are requirements that call for a more human appreciation, such as (R2). Finally, some requirements need to be further refined by domain experts, such as (R5).

[3] They are chosen randomly to illustrate concretely our approach.

4.2 Infrastructure model

Logical model

Let us now focus on the infrastructure itself before undertaking the risk assessment. A synoptic diagram is shown on figure 5. This cloud infrastructure is composed of four zones:

- A *Demilitarized Zone* (DMZ) consisting of public-facing services, such as
  - web servers, for the company's main websites and open banking APIs for its B2B services,
  - authoritative DNS servers, to ensure the proper resolution of the company's domain names,
  - mail servers, for its day-to-day communication needs;
- A regular backend, for most of its business operations, with, among others,
  - databases (primary and secondary LDAP and SQL servers) for its queryable data,
  - business software (customer relationship management and enterprise resource planning software) for its management activities,
  - in-house software for its core processes;
- A PCI DSS backend, for its electronic payment activities, with
  - card processing software,
  - sensitive cardholder and transaction databases;
- A VPN for the network access of its ATMs.

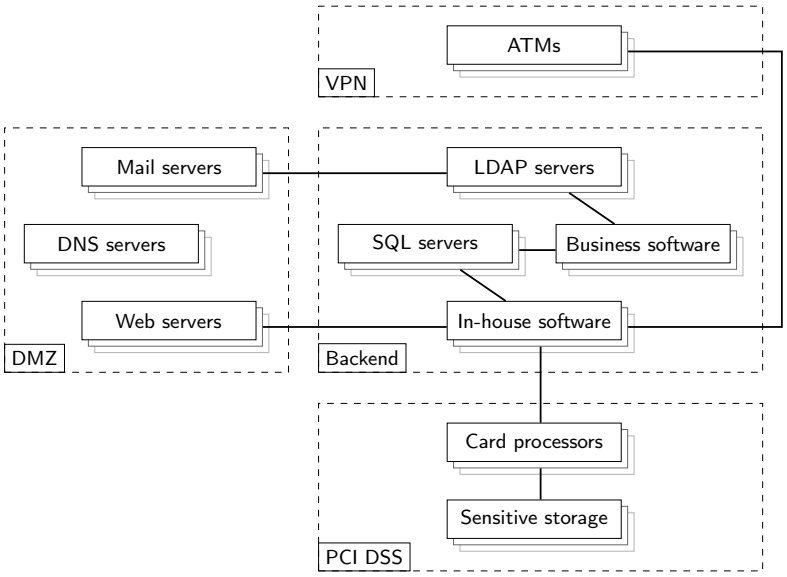


Fig. 5: Infrastructure zones

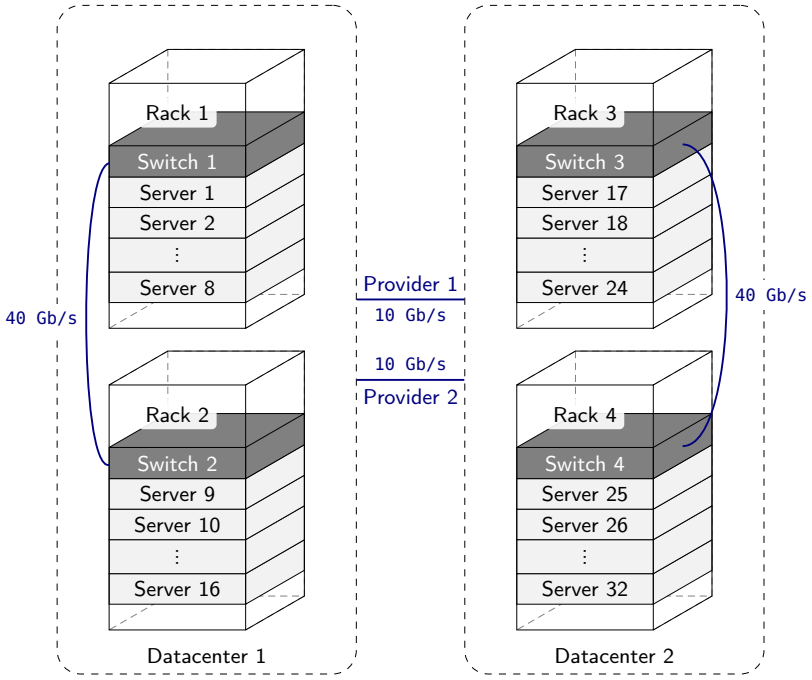


Fig. 6: Physical infrastructure

Physical model

Every cloud infrastructure eventually relies on a physical one. The company owns two datacenters in two different locations and allocated 32 servers (16 in each datacenter) for its private cloud. These servers are distributed in several rooms, as shown on figure 6 and both data centers are connected via two separate network links from two providers.

4.3 Constraints

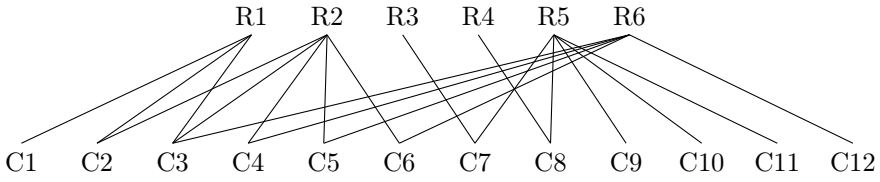
From requirements...

Considering the infrastructure described above, the requirements (R1) to (R6) can be broken down into the following constraints (the terms in small capitals are defined in [7]):

(C1) Users MUST be able to access their personal data;
(C2) Personal information MUST be securely stored;
(C3) Access to personal data MUST be subject to strong authentication;
(C4) Non-PCI DSS server MUST NOT access PCI DSS databases directly;
(C5) Non-PCI DSS VM MUST NOT be collocated with a PCI-DSS one;
(C6) PCI-DSS VMs MUST run on PCI-DSS nodes;
(C7) 99.9 % of transactions MUST be processed within 3 seconds;

(C8) Open banking APIs MUST NOT return HTTP 5XX errors for more than 0.01 % of the transactions;
(C9) Users MUST be able to connect 99.7 % of the time on their web account;
(C10) User incident reports MUST be handled within 12 business hours;
(C11) Primary and secondary database nodes SHOULD NOT be collocated;
(C12) DMZ VM MUST NOT be collocated with backend VMs.

These constraints are linked to the requirements as such:



... to deployment

These constraints impact directly the deployment model of the logical infrastructure on the physical one. For example, from an interpretation of C2, we can derive the following configuration constraints:

(C2α) Disks storing personal data MUST be encrypted;
(C2β) Physical access to servers hosting disks storing personal data MUST be restricted to authorized personnel;
(C2γ) Data flows carrying personal data MUST be encrypted.

As another example, the following constraints can be chosen to provably satisfy (under the hypothesis that the switches have a perfect VLAN isolation) C4:

(C4α) PCI DSS databases MUST be on their own VLAN;
(C4β) Other logical components MUST NOT be on this VLAN;
(C4γ) PCI DSS VMs MAY access the PCI DSS databases VLAN;
(C4δ) Non-PCI DSS VMs MUST NOT access this VLAN.

4.4 Risk

Side-effect analysis

From the constraints, the following hazards can for example be identified :

(C1) — (H1) Personal data can be stolen from users if their machines get infected;
(C1) — (H2) Personal data can be stolen from users if their passwords get stolen;
(C3) — (H3) Personnel losing a physical factor cannot do their work;
(C3) — (H4) Users losing a physical factor cannot access the services in case of an emergency;
(C7) — (H5) The company accepts transactions that should be rejected;
(C10) — (H6) Time constraints put pressure on the employees.

From there, the company needs to update its requirements. (H1) is considered out of the scope of the company and (H2) is already covered by (C3). (H5) is a hazard difficult to balance. On the one hand, in case of a network congestion and during peak hours, transaction times increase and the issuing bank risks paying penalties if the transactions time out. On the other hand, the company may be tempted to automatically accept some transactions, but may then be liable in case of a fraudulent payment, and therefore reimburse the payment out of his own funds. Finally, the company does not have the internal skills to handle (H6) and needs to outsource them.

(H3) —— (R7) The company must deal with lost or stolen authenticators;
(H4) —— (R8) Emergency services must be offered to clients;
(H6) —— (R9) Time pressure on employees must be handled.

The company then has to implement the requirements with constraints and do the recursive side-effect analysis again until it ends. It should be noted that the boundaries between requirements and constraints is blurred. The distinction is not always useful, but we consider here that constraints generate side-effects on risks, while requirements are rules independent of the infrastructure.

Part analysis

The company has deployed a *Configuration Management Database* (CMDB) that provides information about hardware (brands, models, locations), OSes, services and their configurations. The CMDB pulls its information from several domain-specific models continuously updated by hand or automatically. To diagnose hardware-related problems, the company can fetch the relevant manuals from its CMDB, but often needs to contact the manufacturer directly. The incident analysis must therefore be outsourced, as the company does not have a sufficiently broad knowledge on the inner workings of such hardware.

Providing incident (and, to take a proactive approach, risk) analysis tools would be an interesting move from the manufacturer, as it would save time for itself and the company. We also believe that it could give more confidence in the reliability of its products.

Assembly analysis

A common risk assessment referential could also allow the company to reuse and share experience on the interactions of cloud infrastructure components: datacenters, hypervision clusters or distributed software have specific properties that their components do not have. Datacenters present risks for the environment and the population. Clusters behave differently depending on their sizes and the voting algorithms they implement. A piece of distributed software may have sub-optimal performance if one of its nodes is operational but shows a degraded capacity.

An example of a generic referential entry for hypervision clusters that the company could use is shown on table 1. It would ideally be parsable by a computer and should be able to interface with the various infrastructure models.

Component	**Hazard**	**Criteria**	**Likelihood**	···
Hypervision Cluster	CPU cache can leak to a collocated VM	VMs are paravirtualized	Moderate	
		VMs are fully virtualized	Low	
	Resource exhaustion	Over-allocation of resources	Very high	
	Badly distributed database	Two database nodes located on the same physical node	Moderate	···
	Saturation of the internal hypervision network	Two nodes communicating heavily while not located the same physical node	Moderate	
	···			

Table 1: Extract of a risk referential entry for hypervision clusters

4.5 Lessons learned

We illustrate in this example how our approach can help business to perform a more thorough risk analysis. We advocate that, in the spirit of modern software development, such analyses should be shared and be interoperable. In the context of dynamic and increasingly distributed infrastructures, automating IT risk analysis could help ensure greater security and safety, where manual analyses, although potentially more advanced, are more suited to static systems.

5 Conclusion and future work

In this article, we have shown, through an example of cloud infrastructure, how risk analysis on modern IT architectures differs from the domains where it is traditionally used, such as avionics or railway systems. Such analyses cannot be comprehensive because of the dynamic nature of IT infrastructures. Additionally, knowledge is limited by the lack of detailed understanding of several hardware or software elements that make them up. The big picture of a company on the other hand can be obtained through rigorous infrastructure modeling. Federating all these views is not the quest for *the* perfect risk analysis for such infrastructures, but rather the search for *a* better analysis, one that gives the company a sufficient degree of confidence and helps it identify previously unforeseen risk scenarios.

The strategies presented in this article are not yet implemented, because they rely on local risk analyses from manufacturers and developers. The work is already done in part in user manuals, but the lack of a common risk interchange format motivates us to establish a risk referential to start the effort. This is a work in progress. On top of that, we are developing a modeling language for IT infrastructures, in order to link together local infrastructure models and risk analyses for computer-assisted risk management on such infrastructures.

References

1. Reference Model of Open Distributed Processing (RM-ODP), http://rm-odp.net/
2. Abbass, W., Baina, A., Bellafkih, M.: Using EBIOS for risk management in critical information infrastructure. In: 5th World Congress on Information and Communication Technologies. pp. 107–112 (2015). https://doi.org/10.1109/WICT.2015.7489654
3. Agence Nationale de la Sécurité des Systèmes d'Information: EBIOS Risk Manager (2019), https://www.ssi.gouv.fr/uploads/2019/11/anssi-guide-ebios_risk_manager-en-v1.0.pdf
4. Alturkistani, F.M., Emam, A.Z.: A Review of Security Risk Assessment Methods in Cloud Computing. In: New Perspectives in Information Systems and Technologies. vol. 1. Springer International Publishing (2014). https://doi.org/10.1007/978-3-319-05951-8_42
5. Anthes, G.: Open source software no longer optional. Commun. ACM **59**(8) (2016). https://doi.org/10.1145/2949684
6. Baybutt, P.: A critique of the Hazard and Operability (HAZOP) study. Journal of Loss Prevention in the Process Industries **33** (2015). https://doi.org/10.1016/j.jlp.2014.11.010
7. Bradner, S.O.: Key words for use in RFCs to Indicate Requirement Levels. RFC 2119 (1997), https://www.rfc-editor.org/info/rfc2119
8. Cristea, G., Constantinescu, D.: A comparative critical study between FMEA and FTA risk analysis methods. IOP Conference Series: Materials Science and Engineering **252** (2017). https://doi.org/10.1088/1757-899x/252/1/012046
9. van Deursen, A., Klint, P., Visser, J.: Domain-specific languages: An annotated bibliography. SIGPLAN Notices **35**(6) (2000). https://doi.org/10.1145/352029.352035
10. European Parliament and Council of the European Union: General Data Protection Regulation (2016), https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679
11. Golra, F.R., Beugnard, A., Dagnat, F., Guérin, S., Guychard, C.: Addressing Modularity for Heterogeneous Multi-model Systems using Model Federation. In: Companion Proceedings of the 15th International Conference on Modularity (MoMo'2016). ACM (2016). https://doi.org/10.1145/2892664.2892701
12. Hannousse, A., Yahiouche, S.: Securing microservices and microservice architectures: A systematic mapping study. Computer Science Review **41** (2021). https://doi.org/10.1016/j.cosrev.2021.100415
13. He, J., Sun, L.: A Review on SLA-Related Applications in Cloud Computing. In: 2018 1st International Cognitive Cities Conference (IC3) (2018). https://doi.org/10.1109/IC3.2018.00027
14. International Electrotechnical Commission: IEC 61882:2016 — Hazard and operability studies (HAZOP studies) – Application guide (2016), https://webstore.iec.ch/publication/24321
15. International Organization for Standardization: ISO 27005:2018 — Information technology – Security techniques – Information security risk management (2018), https://www.iso.org/standard/75281.html
16. International Organization for Standardization: ISO 31000:2018 — Risk management – Guidelines (2018), https://www.iso.org/standard/65694.html
17. Jackson, D.: Software Abstractions: Logic, Language, and Analysis. The MIT Press, second edn. (2011)

18. Leveson, N.G., Fleming, C.H., Spencer, M., Thomas, J., Wilkinson, C.: Safety assessment of complex, software-intensive systems. SAE International Journal of Aerospace **5**(1) (2012). https://doi.org/10.4271/2012-01-2134
19. Lv, J., Rong, J.: Virtualisation security risk assessment for enterprise cloud services based on stochastic game nets model. IET Information Security **12**(1) (2018). https://doi.org/10.1049/iet-ifs.2017.0038
20. Masoudi, R., Ghaffari, A.: Software defined networks: A survey. Journal of Network and Computer Applications **67** (2016). https://doi.org/10.1016/j.jnca.2016.03.016
21. Mastercard: Transaction Processing Rules (2021), https://www.mastercard.us/content/dam/public/mastercardcom/na/global-site/documents/transaction-processing-rules.pdf
22. Merabti, M., Kennedy, M., Hurst, W.: Critical infrastructure protection: A 21st century challenge. In: 2011 International Conference on Communications and Information Technology (ICCIT) (2011). https://doi.org/10.1109/ICCITECHNOL.2011.5762681
23. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4) (1989). https://doi.org/10.1109/5.24143
24. Neville-Neil, G.: I Unplugged What? Communications of the ACM **65**(2) (2022). https://doi.org/10.1145/3506579
25. OMG: Unified Modeling Language (UML), Version 2.5.1 (2017), https://www.omg.org/spec/UML/2.5.1
26. Payment Card Industry Security Standards Council: Payment Card Industry Data Security Standard (2022), https://www.pcisecuritystandards.org/documents/PCI-DSS-v4_0.pdf
27. Rose, L.C.: Risk Management of COTS Based Systems Development. Springer Berlin Heidelberg (2003). https://doi.org/10.1007/978-3-540-45064-1_16
28. SAE International: ARP4761 — Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment (1996), https://www.sae.org/standards/content/arp4761/
29. SAE International: AS9100D — Quality Management Systems – Requirements for Aviation, Space, and Defense Organizations (2016), https://www.sae.org/standards/content/as9100d/
30. Sulaman, S.M., Beer, A., Felderer, M., Höst, M.: Comparison of the FMEA and STPA safety analysis methods–a case study. Software Quality Journal **27**(1) (2019). https://doi.org/10.1007/s11219-017-9396-0
31. The Open Group: ArchiMate® 3.1 Specification, https://publications.opengroup.org/c197
32. Varga, A., Hornig, R.: An Overview of the OMNeT++ Simulation Environment. In: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops. Simutools '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2008)
33. Visa: Visa Core Rules and Visa Product and Service Rules (2022), https://bb.visa.com/content/dam/VCOM/download/about-visa/visa-rules-public.pdf
34. Yates, A.: A framework for studying mortality arising from critical infrastructure loss. International Journal of Critical Infrastructure Protection **7**(2) (2014). https://doi.org/10.1016/j.ijcip.2014.04.002