On the necessity for high-availability data center backends in a distributed wireless system

Bernd Pfitzinger[1], Tommy Baumann[2], Dragan Macos[3], Thomas Jestädt[1]
[1]Toll Collect GmbH, Linkstr. 4, 10785 Berlin, Germany. Email: bernd.pfitzinger,thomas.jestaedt@toll-collect.de
[2]Andato GmbH & Co. KG, Ehrenbergstr. 11, 98693 Ilmenau, Germany. Email: tommy.baumann@andato.com
[3]Beuth Hochschule für Technik, Luxemburger Str. 10, 13353 Berlin, Germany. dmacos@beuth-hochschule.de

Abstract

When business processes depend on the processing capabilities within a data center, the typical system architecture use a high-availability setup to maintain a high level of service. Faced with a specific machine-to-machine system consisting of many endpoints that collect and forward data to the data center we argue that the dependability of the overall system does not necessitate a high level of service for the data center components. Taking an existing discrete event simulation model of a distributed technical system we investigate and discuss the effects of prolonged outages of the data center on the major business processes of the system.

1. Introduction

The *value chain* [1] perspective emphasizes those activities that contribute directly to the product and the firms' profit. Beyond these primary activities a sizeable number of support activities contribute indirectly. However, one support activity – the day-to-day operations of the information systems – is *de facto* already part of every primary activity: Business processes need information and information systems to run efficiently and to sustain a competitive advantage [2].

In a software-intensive world business processes depend – in the sense of [3] – on the information processing where dependability has five major attributes: availability, reliability, safety, integrity and maintainability. In this article we focus on two attributes, availability and reliability, to discuss their effects on the architecture of information systems.

The starting point for our investigation is an existing distributed machine-to-machine system, the German automatic toll system. This system collects the tolls from heavy goods vehicles (HGVs) driving on federal toll roads in Germany (for more details see e.g. [4] and references therein). About 90% of the tolls are collected automatically by the more than 950000 on-board-units deployed in the HGVs (upper part in figure 1), for the remainder the users choose either a manual log-on at a toll station terminal or the internet log-on to pay the tolls for a planned route in advance. The toll system collects a total of ~ 4.4 bn € annually [5] and is an example for a liability-critical system, i.e. errors may directly result in financial damages.

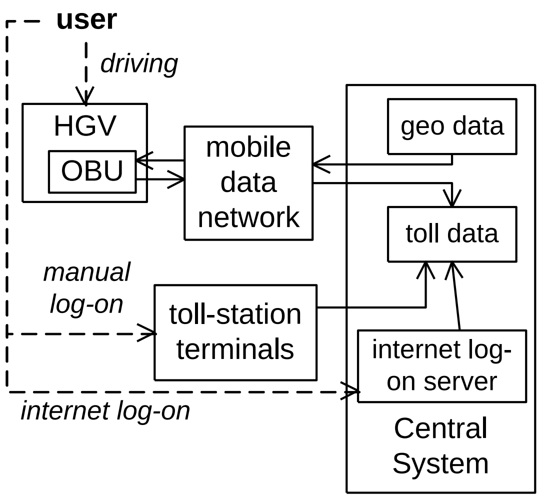

Figure 1: High-level architecture of the toll system with a fully automatic mode using OBUs and a manual log-on via internet or toll station terminals

In the next section we describe the typical steps undertaken to achieve a high level of service – predominantly in the backend systems. Section 3 takes the point of view of the end-to-end processing in the case of the automatic toll collection. We argue that the machine-to-machine system can leverage the distributed clients, i.e. the OBUs in the toll system example, to achieve dependability. To that extent section 4 describes a sequence of simulation runs where the backend systems are offline for increasing periods of time. During the offline period the OBUs buffer the data and start transmitting once the backend systems become available. The accuracy of the simulation results depends (among other things) on the statistics used in modeling the user behavior. Section 5

URI: http://hdl.handle.net/10125/41926
ISBN: 978-0-9981331-0-2

discusses these statistics and compares the simulation model with observations from the real-world system.

2. Achieving a high level of service

The main purpose of a software-intensive system is to render a service – either for a person or to another technical system. In the generic terms of [3] the service is the systems' behavior as perceived by the users in terms of what the system is intended to do. This definition allows for a degree of recursiveness: Figure 1 could be read as showing a single system, i.e. the high-level architecture of the German toll system, or many loosely connected systems interacting with each other.

These opposite views of what constitutes a 'system' lead to different conclusions when discussing non-functional requirements [6], e.g. expressed as constraints (e.g. regarding the interfaces, performance, operations, life-cycle or software economics [7]): Who or what depends on the correct functioning of the system? The generic answer should include a specific requirement, e.g. concerning the quality of the service [8]. Looking back at [3] the quality of the service could be described amongst others in the terms of availability and reliability:

- Is the service supposedly rendered by the system available?
- Does the system provide the correct service continuously?

The latter question is the topic of section 4.2 where we discuss the quality attributes reliability and integrity for the period of resuming normal operations after a prolonged unavailability of the central system. The answer to the first question is usually given as a metric measuring the availability of the service. However, the metric depends on the point-of-view: If the system is seen as a system-of-systems, i.e. each block in figure 1 constitutes a separate system; the availability is defined and measured for each system on its own. This approach readily leads to very high demands on the availability of systems. An example is given in figure 2 where two systems A and B follow the same architecture using a (simplified) layer model. When application B is seen as rendering a service for application A, the availability should be measured at the application-level of system B.

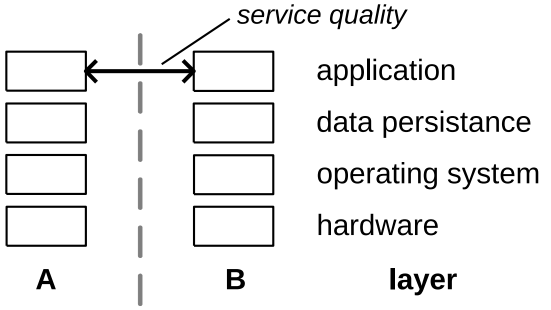

Figure 2: Stylized layer model of two application stacks A and B rendering a service at a given quality.

A simple assumption for the availability of an application stack as depicted in figure 2 is the "*and*" combination of the component availabilities: The hardware with its data center surroundings must be operational, the operating system the data persistence layer and the application itself running to render the service of a single system. One consequence could be to raise the availability of each component: The data center can be built without single-point-of-failures, e.g. up to multiple active power and cooling paths in a Tier-IV data center [9], [10] yielding a site availability of > 99.99%, server and storage hardware with internal error detection and recovery and support staff on-site around the clock. Increasing the availability even further requires the duplication of the system under consideration. When a fault occurs either in the soft- or hardware the system switches to the replica and continues operating. In that way many (but not all) faults can be hidden [11] – increasing the time-to-failure and reducing the time-to-repair.

Increasing the requirements for the availability of each system the cost of operations shows a disproportionate increase. Over the past decade novel IS operators started to offer *cloud computing* where some functionality partakes in existing very large installations with massive redundancy (e.g. [12]). Looking at the toll system example at least some systems offer themselves for a cloud solution, e.g. the internet log-on system or the systems receiving the toll data and providing updates to the OBU fleet. Other systems, e.g. the large-scale billing application, might not yet be ready to transition to a cloud provider – with availability and business continuity listed as a primary obstacle [13].

Returning to the toll system example from figure 1 the emphasis on the availability of sub-systems is at least in parts misleading: These systems are either sourced from business partners or physically deployed in the HGVs. Therefore the next section shifts the focus to discuss availability along the complete business process of collecting tolls.

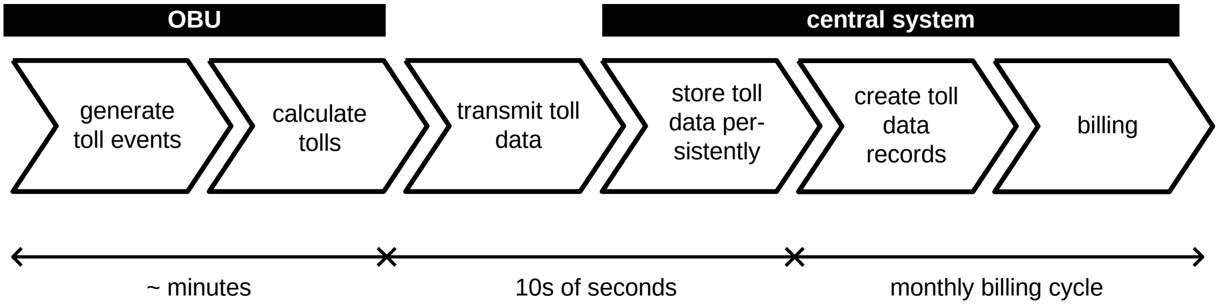

Figure 3: The value chain of the automatic toll system and the typical time scales of automatic processes. The simulation model includes an abstraction of the OBU processes, the data transmission and the servers receiving and storing the toll data.

3. End-to-end processing

Looking at the toll system as a single system rendering the service of collecting the tolls due for driving on German federal toll roads the attribute *availability* could be defined for the end-to-end processing of the tolls. This perspective takes a large part of the value chain (see figure 3) – generating toll data through the use of OBUs, transmission of toll data to the centrals system and download of updates to the OBUs – to define the service rendered to the user, i.e. the HGVs' driver. Apart from power-cycling the OBUs the automatic toll system does not require user interaction, i.e. it is a machine-to-machine system. In that respect we suggest to use the metric POFOD (probability of failure on demand), the proportion of the overall fleet where OBUs are no longer able to generate new toll data. When this happens the OBU signals its unavailability to the user and the user is required to participate in the manual or internet log-on instead. Different reasons can trigger this situation: The local storage space is exhausted, domain-specific limitations are reached (e.g. the maximum time period without successful communication or the credit implied by the stored toll data). These triggers depend only on the usage of a specific OBU – in that sense there is no fleet-wide failure mode, the high-availability is inherent in the usage of independent OBUs. Of course, the possibility of system-wide software faults exists and must be addressed by other means.

As long as the sub-systems are loosely coupled, the components can hide periods of unavailability: Neither the billing process nor the update process is required to run in real-time (for the typical time scales see figure 3, lower part). Similar to telco operations, the billing cycle typically operates on a monthly basis, updates of geo and map data take days to weeks to propagate across the OBU fleet. Toll data is of course generated in direct proportion to the HGV driving at a time scale of seconds or minutes – as toll events while on a toll road in the case of a thick-client OBU or as positional data for subsequent processing in the central system in the case of a thin-client OBU.

Following this approach the metric for the systems' availability is the proportion of the OBUs that are powered on and signal 'out-of-service' to the driver. Anything leading up to this unavailability corresponds at most to delayed processing – toll data needs more time to arrive in the data center and is therefore included only on future bills. It has to be assumed that the operator of the toll system is required to finance the delayed processing, i.e. the typical amount of tolls needs to be transferred to the German federal government in time and can only be recovered from the users when the billing process has caught up with the data backlog. In addition to the cost of financing the operator is assumed to bear the risk of default on the side of its users, i.e. over time some users become unable to pay past tolls.

4. Simulation setup and results

To investigate the attribute *availability* from an end-to-end perspective we use an existing realistic discrete event simulation model of the German automatic toll system [14]. To measure the POFOD metric we enhance the OBU logic in the simulation model to include different triggers for the OBU to signal its unavailability. The most important trigger is the exhaustion of the storage space reserved for toll data – a fleet-wide parameter that we discuss in section 4.1 in more detail. Another trigger in a thick-client

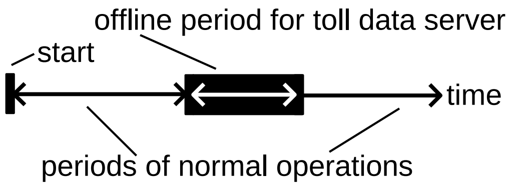
offline period for toll data server
start
time
periods of normal operations

Figure 4: The simulation runs include an offline period of the toll data server with a duration of 1 to 10 consecutive days.

OBU is the validity of the OBUs' geo and map data. In this article we assume that the server providing updates remains available at all times and as a consequence the validity of the local data is a negligible trigger for OBUs to be out-of-service. However, at the end of a fleet-wide update when the new map and geo data becomes valid, those OBUs that were unable to fetch the update in advance will be out-of-service until the update is successfully downloaded and applied.

In addition to the OBU-logic we added the data mining capabilities to the simulation model to measure the fleet-wide percentage of OBUs that are out-of-service in two different ways:

- Counting all OBUs even when they are powered off or not within the reach of the German mobile data networks or
- taking only the active OBUs (powered-on and in Germany) into account.

The latter is close to the POFOD metric, i.e. it is the percentage of OBUs signaling the driver that the OBU is out-of-service. While the driver will perceive this as outage, the real metric would be lower since HGVs spend only ~ 50% of their driving time on toll roads.

The users' perspective expressed with the POFOD metric is an *emergent property* of the whole socio-technical system: "[…] that which cannot be predicted through analysis at any level simpler than that of the system as a whole" [15]. To investigate the effects of server outages on the automatic toll collection we set up a series of simulation runs (see figure 4) where the server receiving toll data is unreachable for 1 to 10 days consecutively starting with the Saturday at the end of the third week in the simulation run. The – seemingly arbitrary – choice of 10 days is motivated by an outage starting on a weekend and running into a week with public holidays (e.g. on Friday and Monday on a Easter weekend in Germany) where the access to personnel would typically be restricted.

Simulation runs for this scenario are repeated for different sizes of the buffer allotted on the OBUs for storing toll data. The simulation runs include another two weeks after the service is restored to observe the return to normal operations. In any other respect we use the existing realistic simulation model without alteration: The size of the OBU fleet is 900 000, OBUs lose their network connectivity briefly and frequently, resource restrictions apply as in the real-world system. Besides the server receiving the toll data no other component has any outages in the simulation run.

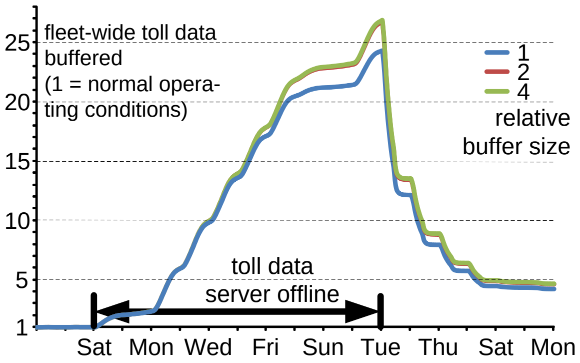

Figure 5: The toll data buffered by the OBUs increases proportionally over time while the receiving toll data server is offline. Choosing a small buffer size leads to an exhaustion (blue line), a larger buffer (red, green) is sufficient for this example.

4.1 Mitigating outages

Under normal operating conditions the OBUs are far from signaling 'out-of-service': Toll data is buffered and there is ample space and time available to hide brief technical outages, e.g. of the mobile data network or the backend system. Map and geo data is distributed in advance so that only those few OBUs that were offline for several weeks need to fetch an update immediately after powering on or returning to Germany. In this scenario OBUs buffer only enough data to allow for efficient technical processes, e.g. considering the server load in the central system and the network characteristics. In effect the buffer holds only a small amount of data under normal operating conditions (normalized as 1 in figure 5).

With the onset of the 10-day period of the toll data server unavailability toll data can no longer be sent from the OBUs to the central system. Over time the OBUs buffer the toll data generated while driving on toll roads (see figure 5), the total amount of toll data buffered increases linearly over time with different rates on weekdays and weekends. Of course, once the

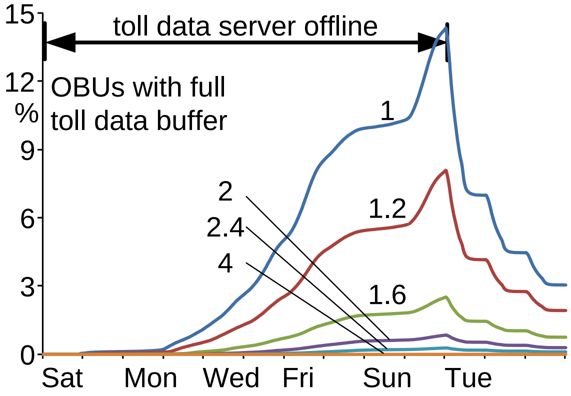

Figure 6: Percentage of OBUs whose toll data buffer is exhausted. 6 simulation results for buffer sizes differing by a factor of 4 are shown.

buffer on a given OBU is exhausted, it can no longer participate in the automatic toll system: In figure 5 the blue line gives the results for a simulation setup where the buffer size is insufficient for long outages. In comparison to buffer size twice or four times as large the total amount of toll data buffered across the whole fleet starts to drop off after five days while the number of OBUs signaling a full buffer increases (not shown in figure 5). At that time some of the most active HGVs covered so many kilometers on toll roads that the buffer is completely full. As time progresses more and more HGVs reach this level. Doubling the buffer size is already sufficient in our scenarios to ensure that almost all OBUs remain in-service over the 10-day offline period.

Taking the users' point-of-view it is important to note that the driving patterns of individual HGVs differ drastically. Almost 10% are either powered off or outside of Germany over long periods of time – buffering toll data is obviously not a concern for these OBUs. Considering the active HGVs the activity follows of course a strong daily and weekly pattern (see e.g. [4]). To determine the buffer size needed to mitigate a given duration of server unavailability we depend on the total amount of toll data gathered by a given OBU over time. The example uses a 10-day period (including two weekends and two days of public holidays) to be buffered by the OBUs – most OBUs cope easily with the outage even when using a small buffer (as shown in figure 5). The most active users quickly exhaust the available buffer space in the first simulation scenario (relative buffer size "1" in figure 6) leading to a sizable portion of the OBU fleet being out-of-service after the 10-day offline-period. It is interesting to note the "saturation" of the buffer across the OBU fleet: The second Monday, i.e. the final day of the offline period, leads to a sudden increase of OBUs out-of-service by about 4 percentage points – the 'average' driving patterns apparently begin to exhaust the buffer after 9 days.

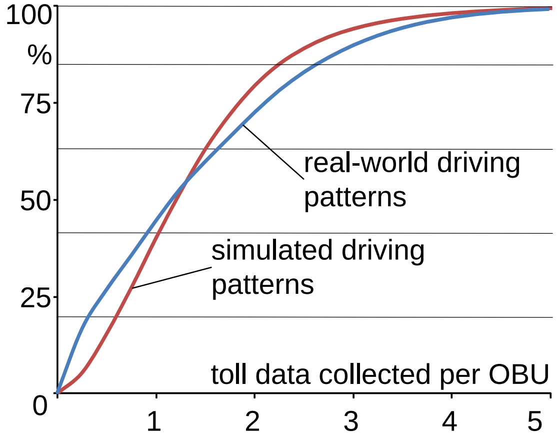

Figure 7: CDF of the toll data collected per OBU over one week as used in the simulation (red line) compared to the real-world driving patterns (blue line).

Increasing the buffer size on the OBUs quickly improves the simulation results (see figure 6). Even a 20% increase in buffer size almost halves the number of OBUs out-of-service after 10 days. Quadrupling the buffer size suffices for almost all driving patterns. Yet some OBUs are very active on toll roads and will still exhaust the buffer. To pinpoint this behavior we look at the cumulative distribution function (CDF) for the amount of toll data generated by an OBU over one week (figure 7, the x-axis gives the toll data in arbitrary units). Over the course of a week most HGVs drive only a moderate distance on toll roads (the median in figure 7 is close to „1“). However, some HGVs are considerably more active – the CDF is shown up to "5" units of toll data collected and still includes only 99,18% of the OBUs in the real-world or 99,38% in the simulation.

Comparing the weekly HGV activity as generated in the driving patterns simulation model (red line, figure 7) with the real-world data (blue line) we note that the driving patterns used in the simulation run deviate noticeably: In reality HGVs generate less toll data than in the simulation. In that way the simulation paints a pessimistic picture of the percentage of OBUs exhausting their buffer space. However, future work is needed to improve the match between the simulated and real-world driving patterns.

4.2 resuming operations

When the server receiving the toll data resumes operation, in our example in the night between Monday and Tuesday after being offline for 10 days, the OBUs quickly reconnect and transmit the buffered toll data. Depending on the retry-mechanism OBUs try to reach the central system periodically with a decreasing frequency so as to avoid any overload situation due to retries. Within the first hours after the toll data server resumes operations most OBUs have completely transmitted their buffered toll data (see the quick fall-off in figures 5 and 6) – if the OBU is powered on and within Germany. Looking at the overall amount of toll data residing on the OBU fleet (figure 5) even several days later almost 5 times as many toll data remain throughout the OBU fleet than prior to the offline period. In turn the risk of losing toll data (e.g. if the OBU is destroyed or the HGVs owner files for bankruptcy) is proportionally higher. The cost of delayed processing is therefore at least the sum of financing the delay and writing off unrecoverable tolls.

All simulation runs show that the server load at the toll data server remains similar to normal operating conditions even when more toll data is transmitted. To gauge the sensitivity of the simulation runs to the server capacity we added one additional scenario where the number of parallel connections is limited to 1/3 of the typical daily peak load. Even then the off-peak hours suffice to return to normal operations with only marginal side-effects.

This work emphasizes the availability of an automatic system and we have shown that end-to-end availability can be assured even if some parts are offline for considerable periods of time. However, one aspect should not be forgotten: [3] mentions reliability as one aspect, i.e. the “continuity of correct service” – in our case the overall detection quota (typically on a level close to 99.9% [17]). Even when we have shown, that the service can resume with almost no impact on users – it did in all scenarios put the OBUs of at least the most active HGVs out-of-service. While these users are still able to log-on manually, the question is open, at what level of impact the system can still be considered to render a “correct service”.

5. Limitations

Dealing with simulations and statistics is in itself a limitation – neither part is perfect and it may not be possible to verify the model with real-world data. Returning to the CDF of the weekly HGV activity we ask whether there is a chance to encounter ‘black swan’ statistics [16]? In some systems – most notably safety-critical systems – even a single extreme event must be considered. In our example the impact on a single user is at most the need to switch to manual log-on or the loss of toll data. What could be the statistical outliers in our case? 0.8% of the OBUs collect more tolls in a week than the maximum value shown in figure 6. This compares to 0.6% as generated by the model of the user behavior used in the simulation runs. In both cases the most active OBUs create about twice as much data as chosen as cut-off in figure 6. Even with the largest buffer used in our simulation runs about 1 in 500 OBUs will be out-of-service as it fills its buffer completely.

The knowledge on the real-world statistics is limited to an analysis of a time period of several months since any older data is and must be deleted. Looking at the maximum daily toll generated by OBUs we see that OBUs can be much more active: A single day suffices to reach well beyond the median of the weekly activity seen in the OBU fleet. If this behavior became the norm even the largest buffer used in our scenarios would fill up within days.

6. Summary

Availability – one of the aspects making up a dependable system – is usually defined in terms of sub-systems, e.g. a server or a network connection. Taking the example of the automatic German toll system we argue that availability in a machine-to-machine system should be defined end-to-end rather than at the component level. Using an existing simulation model we showed that a prolonged offline-period of a server can be mitigated through local buffering of toll data. Looking at the underlying statistical data we note that the generated data differs from the real-world observations, the simulation includes too many highly active OBUs.

7. References

[1] M. E. Porter, Competitive advantage: Creating and sustaining superior performance. Simon and Schuster, 1985.

[2] M. E. Porter and V. E. Millar. How information gives you competitive advantage. Harvard Business Review, 63(4):149-160, 1985.

[3] A. Avizienis, J-C Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing”, IEEE Transactions on Dependable and Secure Computing, pp. 11-33, 2004.

[4] B. Pfitzinger, D. Macos, and T. Jestädt, “Exploring the heavy goods vehicle fleet behavior through simulations: Notes from the German toll system”, IET Intelligent Transport Systems, vol. 9, issue 3, April 2015, pp. 285 – 292.

[5] Bundesministerium der Finanzen, „Haushalts-abschluss 2014“, [accessed 26-May-2015], Jan. 2014. [Online]. Available: https://www.bundesfinanzministerium.de/Content/DE/Monatsberichte/2015/01/Downloads/monatsbericht_2015_01_deutsch.pdf?__blob=publicationFile&v=3

[6] L. Chung and J. C. Sempaio do Prado Leite, “On non-functional requirements in software engineering”, in A. T. Borgida et al. (eds.), “Mylopoulos Festschrift”, LNCS 5600, pp. 363-379, 2009, Springer.

[7] G.-C. Roman, “A taxonomy of current issues in requirements engineering”, Computer, vol. 18, issue 4, pp. 14-23, April 1985.

[8] M. Glinz, “On non-functional requirements”, 15th IEEE International Requirements Engineering Conference, pp. 21-26, 2007.

[9] P. W. Turner, J. H. Seader, and K. G. Brill, “Tier classification define site infrastructure performance”, Uptime Institute, vol. 17, 2006.

[10] M. Wiboonrat, “System reliability of fault tolerant data center", in CTRQ 2012, The Fifth International Conference on Communication Theory, Reliability, and Quality of Service, 2012, pp. 19-25.

[11] K. Trivedi, G. Ciardo, B. Dasarathy, M. Grottke, R. Matias, A. Rindos, and B. Vashaw, “Achieving and Assuring High Availability”, in “Service Availability”, Lecture Notes in Computer Science, T. Nanya, F. Maruyama, A. Pataricza, and M. Malek, Eds., vol. 5017, Springer Berlin, 2008, pp. 20-25.

[12] L. A. Barroso and U. Hölzle, “The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines", Synthesis Lectures on Computer Architecture, vol. 4, no. 1, 2009.

[13] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A View of Cloud Computing", Communications of the ACM, vol. 53, no. 4, pp. 50-58, 2010.

[14] B. Pfitzinger, T. Baumann, D. Macos, and T. Jestädt, “Modeling regional reliability of 2G, 3G, and 4G mobile data networks and its effect on the German automatic tolling system", in 2015 48th Hawaii International Conference on System Sciences (HICSS), Jan. 2015, pp. 5439-5445.

[15] G. B. Dyson, “Darwin among the machines: The evolution of global intelligence”, Basic Books, 1998.

[16] N. N. Taleb, “The black swan”, Random House, 2010.

[17] M. Dettmar, F. Rottinger, and T. Jestädt, “Achieving excellence in GNSS based tolling using the example of the German HGV tolling system”, ITS Europe, Dublin, 2013