Design Criteria and Design Concepts for an Integrated Management Platform of IT Infrastructure Metrics

Christian Straube, Wolfgang Hommel, and Dieter Kranzlmüller
Munich Network Management Team, Ludwig-Maximilians-Universität München, Leibniz Supercomputing Centre
[*straube,hommel,kranzlmueller*]@*mnm-team.org*

***Abstract*—Most measurement and metrics as they are used in today's IT management are not suitable for profound upper level management decisions. We identify four major gaps between the information that can be measured on a technical level and the information that is needed for management decision making: 1) The currently provided information is not suitable for decision making on higher abstraction levels. 2) Interdependencies between the metrics are not sufficiently considered. 3) There is no support for the derivation of improvement recommendations based on the metrics values. 4) Existing approaches lack the flexibility to incorporate organization-specific requirements. Based on state-of-the-art energy efficiency, performance, and security metrics taken from related work, we present how these gaps affect a complex real-world scenario. Consequently, we argue that an integrated management approach for IT infrastructure metrics is necessary and present the core components of our solution, referred to as the *management cockpit*. We therefore discuss its four-layered architecture, which deals with measurements and metrics, dependency handling, aggregation logic, and graphical representation as well as its information model backend. Finally, we present an overview of related work and give an outlook to open issues and future work.**

***Keywords*-Decision making support, measurements and metrics, integrated IT management, management tools, energy efficiency management, IT service management, service reporting**

I. Introduction

Contemporary *Information Technology* (IT) infrastructures are highly complex and mostly distributed artifacts, forming several specialized groups, like supercomputers, clusters, Grids or enterprise IT infrastructures. The ongoing development in the related areas, the by now short improvement cycles of the employed hardware and software, manifold business needs and the ever-changing environment of IT infrastructures, like changing customer demands or legal aspects, require a continuous adaption of the entire IT infrastructure and its sub parts. For instance, new security threats must be addressed, faster hardware is required to tackle strong competitors, or existing hardware turns out to be error prone.

Adapting an IT infrastructure to the above outlined situation and aligning it to the current requirements can be achieved by changes and modifications to the hardware, the software, and the configuration of an IT infrastructure. For instance, adapting CPU frequency to achieve a power consumption decrease or introduce redundancy to improve reliability are typical modifications. Each of them, however, must be thoroughly planned for two reasons [1]. First, planning is required to address the mostly business-critical dependency on IT infrastructures in nearly all areas, since IT infrastructures provide the foundation of IT services and IT-based business initiatives [2], [3], [4] a whole enterprise might depend on [3]. Second, planning should circumvent unnecessary adaptations and avoid their (mostly) costly investigation. A decision to purchase new hard disk drives for an entire cluster to improve reliability, for instance, requires a study covering potential manufactures as well as investigating the interoperability with the existing hardware. If it turns out that other components were much more unreliable than the replaced hard disk drives, the study (*investigation*) and the replacement efforts (*change*) were wrongly placed.

The important role of IT infrastructures for science and industry, the (potentially) severe impact of IT infrastructure changes to an enterprise's success and the power of changes to align an IT infrastructure to its surrounding cause an involvement of (upper and top-level) management in nearly every change planning and decision making process, especially strategic, large-scale, and cost-intensive decisions.

Sustainable decisions about the continuous improvement and future development of an organization's IT infrastructure should be based upon solid knowledge and database about an IT infrastructure's current state. The big importance of information for the decision making process has been investigated by current and comprehensive studies, e.g., "Big data Harnessing a Game-Changing Asset" conduced in 2011 by the *Economist Intelligence Unit* [5]. One of its key results: 90% of the decisions made within the last three years would have been significantly better, if (more) relevant information would had been available. Even if this study focused on retail and financial data, this is, in principle, also true for information about IT infrastructures.

Achieving *objective*, *transparent*, and *quantitative* decisions poses several requirements on the above mentioned solid database and its creation and maintenance, i.e., incorporating *measurement values* of selected characteristics and the employment of carefully chosen *metrics*. Over the past few years, multiple metrics have been specified by re-

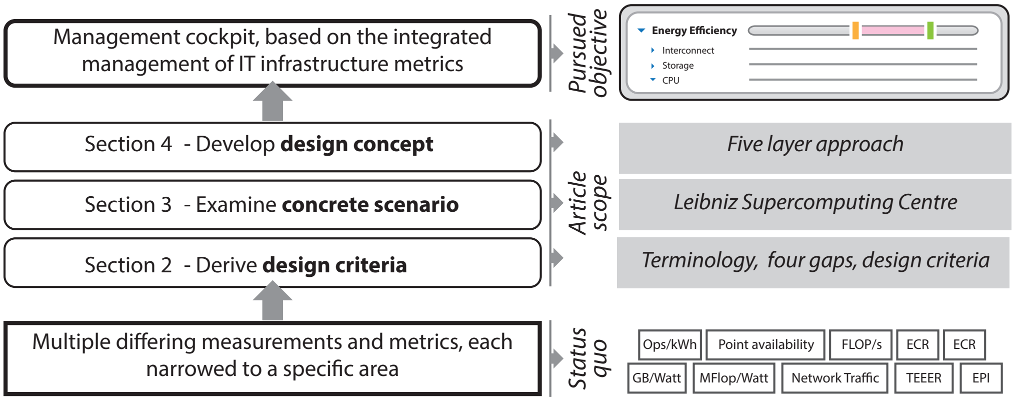
Management cockpit, based on the integrated management of IT infrastructure metrics
Pursued objective
Energy Efficiency
Interconnect
Storage
CPU
Section 4 - Develop **design concept**
Section 3 - Examine **concrete scenario**
Section 2 - Derive **design criteria**
Article scope
Five layer approach
Leibniz Supercomputing Centre
Terminology, four gaps, design criteria
Multiple differing measurements and metrics, each narrowed to a specific area
Status quo
Ops/kWh
Point availability
FLOP/s
ECR
ECR
GB/Watt
MFlop/Watt
Network Traffic
TEEER
EPI

Figure 1. Scope of the article, i.e., the process of deriving requirements from a concrete (exemplary) scenario, resulting in a design concept for an integrated management platform of IT infrastructure metrics

searchers and practitioners for several IT management areas, including, among others, quality of service, performance, energy efficiency, and information security. Unfortunately, only few of them have been standardized, and most literature and real-world implementations focus on only a single one of these metric categories. For instance, metrics regarding energy efficiency on the one hand and information security on the other hand are rarely discussed by the same group of technical experts as of today.

In this context, a solid database for decision support can be achieved by an *integrated IT infrastructure metrics management*, emphasizing the holistic manner and exhaustive consideration of an IT infrastructure. The development of an integrated IT infrastructure metrics management faces several challenges and problems, especially when it comes to supporting high-level management decisions, in particular

1) the insufficiency of information provided by (low-level) metrics for decisions on higher levels,
2) the dependencies between measurements and metrics,
3) the incorporation of environmental influencing factors,
4) the lack of activity recommendations that can be deduced from metric values, and
5) the plethora of existing measurements and metrics at a lower abstraction level or rather at hardware-level.

This article presents a design concept for a *management cockpit* that addresses the above itemized challenges and aims at supporting top-level management decisions about planned changes based on a profound knowledge and database applying an integrated management of IT infrastructure metrics. In particular, the design concept describes a possible way to split the aforementioned extensive development problem in smaller parts and an architecture of implementation building blocks and their interactions. Additionally, the design concept describes guidelines and advice how to implement a particular part. In contrast, the design concept presented in this article does not provide concrete low-level implementations, e.g., a concrete aggregation rule. This is on purpose, since the concrete implementation of the outlined building blocks heavily depends on a variety of factors, like the objectives of the top-level management or the existing measurements and metrics.

In short-term, the management cockpit allows the visualization of raw measurements as well as derived metrics and shows their interdependencies to facilitate profound decision making. In long-term, it can also be used as a simulation tool for planning, optimizing, and choosing between mutually exclusive alternative options. This is of special importance for management decisions regarding investments in hardware, since unlike for the tweaking of software parameters, there usually is no viable rollback plan for hardware changes, such as upgrading CPUs or replacing HDDs, if it turns out that the performed change did not bring the desired effects.

Figure 1 depicts the article's scope and the discussed elements in the context of developing the outlined management cockpit. On the left hand side, Figure 1 shows necessary steps and section structure, on the right hand side it provides some detailing information. Since a systematic development process starts with analyzing *requirements* [6], [7], the process begins with a thorough consideration of the current situation, i.e., "multiple differing measurements and metrics, each narrowed to a specific area", depicted at the bottom of Figure 1. The findings are summarized in Section II in a set of design criteria, consisting of a basic terminology, four identified gaps, and some gap-spanning design criteria. To further substantiate the discussed design criteria and to provide a tangible example, Section III examines a concrete real-world scenario, the *Leibniz Supercomputing centre* (LRZ) and its High Performance Computing (HPC) system SuperMUC. Section IV presents a layered design concept to accomplish the above outlined management cockpit in general and to close the identified gaps in particular. It presents, among other topics, our information model for metrics, dealing with their inter-dependencies and visualization challenges. Section V discusses related work that has influenced our design; finally, Section VI concludes this article with a summary and an outlook to our next steps.

II. Gap analysis and design criteria

This section provides a *fundamental terminology*, identifies *four gaps*, and outlines gap-spanning *design-criteria*.

The fundamental terminology in Section II-A aims at ensuring a common understanding of important terms and concepts in the context of this article. This is required by the overload and differing meanings in different areas, like *metric* or *IT infrastructure*. Section II-B identifies a set of gaps that have to be filled or solved by the management cockpit development. The subsequent Section II-C outlines gap-spanning design criteria that have to be addressed during development, e.g., criteria for high quality measurements. Both, the four gaps and design criteria, act as *requirements* that start a systematic development process [6], [7]. For simplicity, gaps and design-criteria are derived and presented in a non-formalized way, e.g., without formulating *use cases* (cf. Jacobson [8]) and hence, the specific term requirement is omitted.

The four gaps and design criteria are substantiated for illustrative purposes in Section III, which examines the LRZ and SuperMUC.

A. Terminology

Almost all important terms used in the context of this article suffer from an overload. For instance, there are several definitions of *IT infrastructure* covering several focal points and granularity levels (cf. [9], [10]) but no definition is commonly accepted as standard or widely applied [11]. This is also valid for terms that are used in the context of assessing, characterizing, and valuating IT infrastructures, e.g., *metric*, *measurement*, or *key performance indicator (KPI)*. Despite a mass of literature about these terms (e.g., [12], [13]) there is no commonly accepted definition yet.

To address this situation and to avoid the risk of comparing and aggregating values with different meanings and intentions, the subsequent itemization provides a set of non-formal definitions. It is supported by Figure 2 and Figure 3, addressing *IT infrastructure* as well as *measurement* and *metric*, respectively. In contrast, we do not target a universal definition.

IT infrastructure – As motivated above, the article does not aim at developing or providing a long-term definition but a common terminology for the design concept presented here. Hence, we apply a very generic definition of *IT infrastructure* to cover as much situations as possible. In particular, the term "infrastructure" is composed of "infra" (lat. "beneath", "under") and "structure", and can be interpreted as "beneath the structure" [11, p. 36]. Despite the focus on information technology implied by the prefix "IT", *IT infrastructure* still might contain elements that are not considered, i.e., non-technical aspects [14], [10], like knowledge, skill-sets or *IT service management* (ITSM) processes, which can be summarized to the "human IT infrastructure" [4]. Furthermore, "IT structure" is interpreted as "IT system", i.e., "a set of things working together as parts of a mechanism or an interconnecting network" [15].

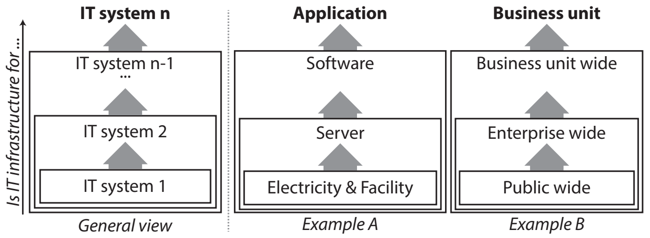

Figure 2. The term "IT infrastructure" is defined relatively to the applying or consumint IT system.

Summarized, *an IT infrastructure is the set of hardware resources that are necessary for running and using an IT system. Relations between set elements are built by functional dependencies and interactions*. Figure 2 illustrates the understanding relative to a given IT system: an IT system can be both, a consuming system and an IT infrastructure at the same time. On the left hand side, the generic layered pattern is depicted, on the right hand side two examples are given. Example A (taken from [11, p. 36]) emphasizes that an IT system can be both, consuming an IT infrastructure and providing an IT infrastructure at the same time. Example B (taken from [16, Figure 1]) illustrates the deployment at multiple levels. Especially example B endorses the relative understanding of IT infrastructure and that *the* one IT infrastructure can not exist and a relative view is mandatory [4].

Component type – Obviously, an IT infrastructure consists of manifold differing components. Enabling the assignment and selection of measurements and metrics requires a distinction of component types. For this, a component type is defined by a component's *capability*, i.e., a well-defined low-level functionality, like computation, data storage, and data transfer, that is exposed to a user or application [17]. Hence, exemplary component types are "data transfer" or "storage". These types can be extended and defined individually for a particular scenario.

Measurement – The considered (real world) objects own an arbitrary set of characteristics that describe the object, summarized as *facts* on the left hand side of Figure 3. In order to enable a reasonable processing of the extensive set of facts, a *measurement* abstracts these facts by reducing information complexity [18], [19] and mapping the resulting remaining facts onto a symbol set, which enables the execution of mathematical functions [20] (cf. Figure 3). Measurement results in a set of *measurement values*, depicted at the middle of Figure 3. There are three types of measurement values, i.e., *simple* values – directly compiled by a measurement, *additive* – compiled by adding two or more simple measurement values, and *derived* – compiled by applying a more complex operation on a set of simple or additive measurement values.

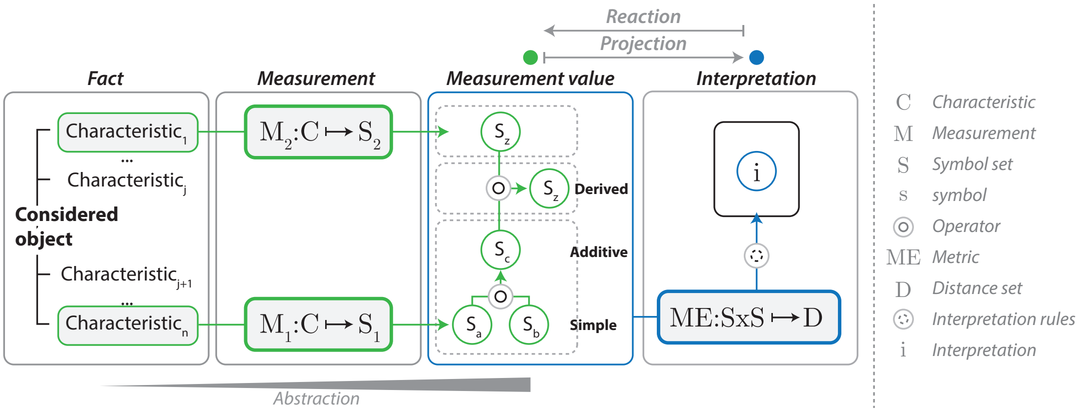

Figure 3. Process from facts that describe a real-world object to an interpretation that is used for top-level management decision making.

Metric – As depicted in Figure 3 on the right hand side, a metric is a distance function, i.e., a function that maps two measurement values $S \times S$ onto a (numerical) distance DV. A metric is required to satisfy four conditions, i.e., *non-negativity*, *identity of indiscernible*, *symmetry*, and *triangle inequality* [18]. In contrast to the (optimally) objective nature of a measurement, a metric creates correlations and evaluation.

Interpretation – An interpretation finally defines the semantics that are relevant to the top-level management decisions. In particular, an interpretation consists of a metric and a finite set of interpretation rules that can be applied on the compiled distance. Consequently, an interpretation evaluates correlations between measurement values. For instance, the distance "20 Bit/second" is enhanced by a good/bad assessment.

Key Performance Indicator (KPI) – A KPI is a specifically selected interpretation of subjective importance.

B. Metrics-based management decision-making gaps

This section identifies and discusses gaps that have to be filled or solved by the management cockpit development. Figure 4 overviews them as well as their general correlations and context. At the bottom of Figure 4 are the existing low-level measurements and metrics, exemplarily represented by some (arbitrarily) selected metrics that consider the energy efficiency of interconnect, storage, and CPU hardware types of HPC IT systems. The aggregation of these low-level metrics and their enhancement for top-level decision support is covered by *Gap 1 – The information gap*. The handling of potential correlations and dependencies between two or multiple metrics is covered by *Gap 2 – The dependency gap*. To further support top-level management in the decision-making process, not only a profound information base is required, but also the (automatic) derivation of recommendations how the top-level management should react on aggregated values and how modifications should be executed and achieved. The thereby arising challenges are covered by *Gap 3 – The activity gap*. Furthermore, the implications and influencing factors caused by the considered IT infrastructure's surrounding range from formal aspects like national law to immutable electricity prices or the housing building. All these elements are covered by *Gap 4 – The environment gap*.

The gaps are subsequently detailed further.

Gap 1 – Information Gap – This gap can be considered as the main gap, as it reflects the fact that information from existing low-level approaches are not available at higher abstraction levels, where it would be necessary to create a comprehensive holistic view. Without the information from low-level approaches, incomplete and incorrect information has to be used to make strategic decisions, e.g., decisions regarding the energy efficiency of the IT infrastructure for a procurement decision. This gap becomes even more severe in the context of the commonly accepted management principle that an activity cannot be managed if it is not measurable [13].

The information gap describes the transformation from the existing measurement and metric values to a (small) set of aggregated values for the top-level management. Available information often is unsuitable for the target audience. For instance, hardly any top manager will be fond of making a decision about which new server CPU hardware to invest in given a metric such as *MegaFlops per Watt*, even if this is an interesting energy efficiency metric for a technician. For a holistic view, the (purely) technical information has only limited expressiveness and must be enriched by context and comparison information.

The information gap covers two questions, i.e., *what* should *how* be aggregated. Additionally, all this information has to be aggregated to provide comprehensive information to support decision making at high level. Therefore, conversions, e. g., into currencies or hours of work, may be required.

The first question about the **what** addresses the selection of a set of metrics for aggregation. Usually there are several measurements and metrics, each focusing on a (slightly) different aspect. Hence, in a very first step it must be

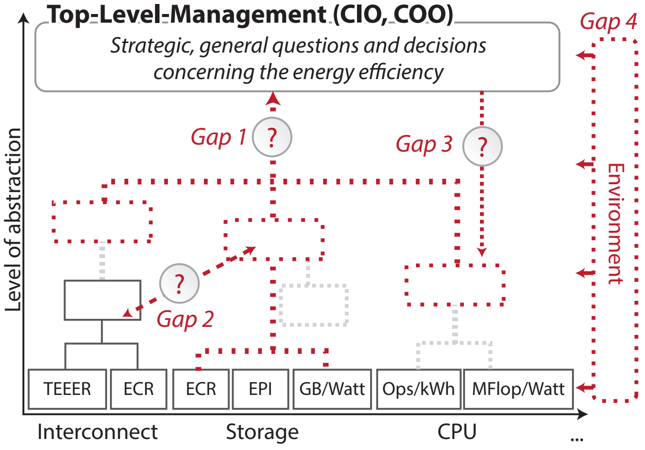

Figure 4. The four gaps in today's situation that hamper a holistic view.

decided, which existing values should be aggregated, which metrics are or could be relevant for the currently considered question, and if each value should be respected with the same priority.

The second question about the **how** covers the differing measurement and metric scales as well as the aggregation function to apply. Not all metrics can be applied to the same component types (cf. Section II-A). For instance, a CPU/GPU metric like *MFlops/Watt* cannot be applied to storage, interconnect, or software components.

Gap 2 – Dependency Gap – The second gap addresses misleading results and interpretation if measurements and/or metrics are considered separately without respecting dependencies and hence, without including all relevant information. Additionally, it mostly is not adequate to improve only one or a small set of metrics, i.e., partial optimization does not yield optimum results. Instead, all (involved) metrics should be improved [21], correlations have to be covered, and conclusions have to be drawn from these correlations [22]. Besides the big advantage of respecting the reciprocity of metrics, considering metric dependencies facilitates the uncovering of strategic goal conflicts [23] and the consideration of trade-offs, for instance, between energy efficiency and performance. Those trade-offs are already analyzed in some lower-level-approaches, e.g., by Rossi et al. [24], but not yet at upper-level management.

Dependencies can be split in *intra*- and *inter*-metric dependencies. **Intra**-metric dependencies address dependencies within a single metric group. In the realm of performance, for instance, an IT infrastructure is determined not only by the computing cores, but also by the communication, interconnect, and I/O performance [25], [26], [27]. **Inter**-metric dependencies cover dependencies between metrics of different groups. For instance, increasing hot-standby redundancy to address short-time breakdown and to improve reliability [28], simultaneously increases energy consumption and degrades performance due to redundancy overhead [29]. Another example is the trade-off between a new type of network component might have a better energy efficiency, but worse performance than a competitive product. Both products may also have different security properties and investment cost.

Gap 3 – The Activity Gap – The provision of a profound knowledge and database to the top-level management as motivated in Section I is covered by the already discussed Gap 1 and Gap 2. Both represent a big step towards a management cockpit based on the integrated management of IT infrastructure metrics. Nevertheless, the values resulting from metric selection and aggregation mostly require a focused activity from top-level management. Hence, Gap 3 questions how to react on certain aggregated values and implications. In other words, in order to perform the best adoptions on the analyzed IT infrastructure, activity recommendations have to be generated out of the holistic view in a (semi-) automated way. It is not sufficient to define or use a set of metrics in order to evaluate a situation. Instead, there are additional information required to assess the impact of a particular value [21]. Figure 3 depicts this as *interpretation* that triggers a feedback mechanism [19].

Recommendation compilation faces two challenges, i.e., *cost optimization* and *metric reciprocity*.

As outlined in Section I, each modification or rather its execution causes a variety of cost, ranging from preparatory investigation to implementation related cost. Additionally, there are mostly at least two potential reactions on an aggregated value and its implications. For instance, an alarming low reliability could be addressed by either introducing redundancy or by replacing the unreliable elements. The caused costs, respectively, have to be considered during recommendation compilation to achieve results as good as possible.

The complexity of contemporary IT infrastructures renders the separation of the specific contribution of a particular infrastructure component very difficult [30], [31]. Additionally, effects induced by local modifications on a single component can quickly and easily cascade and affect the HPC infrastructure partly or completely [32]. Activity recommendation compilation is required to consider effect cascading in order to avoid unpredictable issues.

Gap 4 – The Environment Gap – This gap is orthogonal to the different abstraction levels and metric dependencies discussed in the last three gaps as Figure 4 illustrates, and hence, the environment gap affects all other gaps. In particular, the environment gap affects *metric semantics* and covers *external factors* that might influence measurements and compiled values as well as factors that (obligatorily) have to be addressed or covered in an abstraction level spanning way.

The **metric semantics** states that the same measurements and metrics may not have the same expressiveness and purpose in different scenarios. Each organization might have to make specific adoptions to existing metrics, create complementary metrics for its specific environment, and specify, for example, how results must be interpreted properly. This is of special importance to the activity gap (cf. above), since result interpretation and implications might have tremendous impact on the taken actions and made decisions. For instance, energy consumption figures might be alarming in a location facing high electricity prices, whereas the same numbers might have low or no importance in a location benefiting from low electricity prices.

The **external factors** cover a diversity of elements, like the building that houses the IT infrastructure or the national law, which guides a company's compliance. The integration of these external factors heavily depends on the specific objectives of top-level management. For instance, if a holistic energy efficiency investigation is pursued, the *Power Distribution Unit* (PDU) must be covered as well as the building and supporting infrastructure, like cooling.

C. Gap-spanning design criteria

There are some important aspects that affect all mentioned gaps and that can be considered as the *non-formal* requirements to fulfill.

Not delimited set of metric groups – Closing the above detailed gaps is already a non-trivial task for the exemplary used areas energy efficiency, performance, and information security. Nevertheless, the integrated metric management and management cockpit must be capable of integrating or at least be extendable to an arbitrary set of measurements and metrics. Many more IT infrastructure capabilities can be harvested for various types of metrics, such as cost, reliability, re-usability, and degree of standardization. It should, however, be kept in mind, that the complexity of the concepts discussed in this article increase with the applied breadth of the capability spectrum.

Allowing multiple-perspective scenarios – Mostly, top-level management consists of several experts from different areas. Hence, the management cockpit should support "multiple-perspective scenarios", i.e., "many different narratives about the same events, with the intention being to explore how the different perspectives might be coordinated or might reach some accommodation" [33]. For instance, there is a user type *client category A*, a business unit *highly critical storage services* or a topic *Urgent Computing*. Each view defines its specific obliged values and objectives, which are in turn considered while planning strategic actions or behavioral guidelines concerning the IT infrastructure. In summary, multiple views shall be consolidated into one holistic view [33] and strategic goal conflicts between perspectives are exposed.

Integrating measurement and metrics data – Supporting the severally motivated holistic view and root cause analysis requires the integration, i.e., the "selection, embedding, and handling of the underlying data sources" [34] and the use of as many data sources as possible. This in turn calls for the consolidation of several data structures and the identification of a valid data context [22]. Enabling the use of the management cockpit from the first day and needs to avoid the "cold-start–problem" [35], and existing and actual data, measurements, and metrics have to be embedded [22].

III. Scenario Leibniz Supercomputing Centre

The complexity of working with a large number of measurements and metrics is easier to grasp when a real-world example is used. Hence, this section illuminates a concrete scenario in order to make this variety more tangible and to provide some examples for the high level of abstraction of the above discussed challenges and issues.

Smaller IT infrastructures, such as small number of servers operated by a university computer science chair or a very small enterprise do not exhibit the same IT management decision problems as large IT infrastructures. Similarly, when research is focused on only a single metric category, the issues resulting from interdependencies we have to face in real-world scenarios are often neglected. Having said that, we use the LRZ as an example because we know the set of problems there in-depth based on several projects and the IT service operations we are involved in.

LRZ is located in southern Germany and has a twofold mission. On the one hand, it is the common IT service provider of all higher education institutions in the greater Munich area. It offers several dozen IT services for more than 130.000 students, faculty, and staff; for this purpose, it operates a four-digit number of server machines and a communication network infrastructure consisting of more than a dozen *Internet Protocol* (IP) routers and about 1.500 network switches, making it well-comparable with larger enterprises. On the other hand, LRZ is one of Germany's largest scientific HPC sites. Besides a large Linux cluster with about 10.000 CPU cores, it operates a supercomputer named SuperMUC, which entered the Top 500 HPC list at place 4 in June 2012 and was Europe's fastest supercomputer. LRZ had to construct a completely new building for SuperMUC, which uses hot liquid cooling, and has received a national award for the energy efficiency of its infrastructure in 2012.

SuperMUC's architecture and relevant characteristics are subsequently outlined in Section III-A. Section III-B then focuses on the selected metrics in general and concrete examples for the SuperMUC in particular. As a core issue, let us assume the following questions that LRZ's management wants a profound answer for: "How can future HPC systems at LRZ be made even more energy-efficient without impacting their performance and scrutinizing their security? Can some of these measure even be already applied to SuperMUC without exceeding a given yearly maintenance budget?"

A. LRZ's and SuperMUC's IT infrastructure

In a *High Performance Computing* (HPC) system there typically are dedicated worker/compute nodes, storage components, a head node and an interconnecting high-bandwidth network [36]. These components interact and exchange information to expose HPC capabilities. Additionally, there are several metrics and measurement approaches for multiple areas, such as availability, performance, or quality of service. Additionally, there are manifold approaches within a single area, e.g., ranging from low-level considerations like investigating the power consumption of an Intel PXA255 processor [37], to high-level considerations, like investigating the power consumption behavior in an actual data centre [38].

Figure 5 depicts a schematic view of SuperMUC's architecture: SuperMUC's compute elements are built of 18 identical *IBM System x iDataPlex* thin node islands. An island comprises 512 nodes, each employing two *Sandy Bridge-EP Intel Xeon E5-2680 8C* processors having 8 cores each, resulting in 147.456 cores. There is an additional fat node island with 40 cores per node and 6.4 GB RAM per core, providing additional 8.200 cores.

Storage elements are split in three areas accordant to their intention. The temporary disk storage for compute job execution is run in IBM's *General Parallel File System* (GPFS), a high-performance clustered file system. The permanent storage, e.g., for `home` directories, are located on a *Network Attached Storage* (NAS) based disk storage.

As depicted in Figure 5, also the network is split in different areas and employs different technologies. Islands and their nodes as well as the temporary disk storage are connected via an *Infiniband interconnect*. The Infiniband interconnect is operated at a *Fourteen Data Rate* (FDR)-10. SuperMUC's size requires the employment of several switches, in particular 20 big island switches and several smaller switches within an island. The archive and backup system is connected via a slower 10 Gb Ethernet.

Additionally, the campus on which LRZ building resides is one of the major backbone sites of the networking infrastructure referred to as the Munich Scientific Network, and provides a 23.5 GBit/s uplink to German's national research and education network, X-WiN. Because LRZ also operates several thousand Linux and Windows servers, NAS filers, and a tape backup and archival infrastructure, several hundred edge and access network switches are used in the LRZ building. In total, more than 450 kilometres of copper and glass fiber cables are used in the single data centre building to provide the required connectivity with a carefully crafted redundancy for high availability that covers technical failures as well as major incidents such as room-local fires.

B. Exemplary metric categories for use at LRZ and SuperMUC

Out of the variety of potential metric categories, we further detail *energy efficiency* (Section III-B1), *performance* (Section III-B2) and *information security* (Section III-B3). The former two are considered the be among the most important ones accordant to the PRACE scientific case [39]. The latter one is an essential area of responsibility for both system administrators and management. Unfortunately – and directly related to how management decisions are made due to the identified gaps – security is not yet in the core focus of most HPC installations. However, as security often requires a trade-off with other goals, such as performance, intertwining all three metrics categories can be expected to become more important in the future.

For each metric category, a general discussion is succeeded by a concrete consideration in the context of the above outlined scenario.

1) Energy efficiency: EE is a severe problem given the background of expected consumption levels of hundreds of megawatts in the future [40], [41] and steadily increasing electricity prices. For many data centres and other IT service providers, raising energy consumption costs are the primary motivation for an in-depth examination of EE technology. EE obviously is important to consider before hardware investments are made; for example, buying new servers with CPUs supporting frequency scaling helps to level energy costs with the current workload throughout the lifetime of the server machines. Buying cheaper servers and replacing the CPUs afterwards typically would lead to a much higher total cost of ownership. However, EE capabilities need to be constantly monitored and several EE parameters need to be dynamically re-configured. For example, air-conditioning for the servers typically needs to be adjusted to environmental characteristics such as the current outdoor temperature.

EE obviously is of utmost importance also for LRZ: SuperMUC consumes about 3 MegaWatts of power when it is under full load, leading to multi-million Euros power cost per year for this single system. SuperMUC's EE therefore clearly dominates LRZ's power bill, but several thousands of other server machines and network components must not be neglected either. For example, state-of-the-art network switches by well-known international vendors differ by factor 2 regarding their waste heat production when power-over-ethernet-enabled models are concerned. This does not only influence the power consumption of the IT equipment itself, but also has consequences for the climate / re-cooling infrastructure because cooling airflows need to be increased.

We now give some examples for the gaps identified in Section II related to EE. The same gaps exist for the metrics categories discussed below but are omitted there for brevity.

Example for Gap 1 – In order to answer the management question how SuperMUC's EE could be further improved, we first have to decide, which components have the poorest energy efficiency in SuperMUC at the moment, as their potential for further improvement during the next system extension is the highest. Besides a few generally applicable metrics, most metrics can be applied only in one area, for

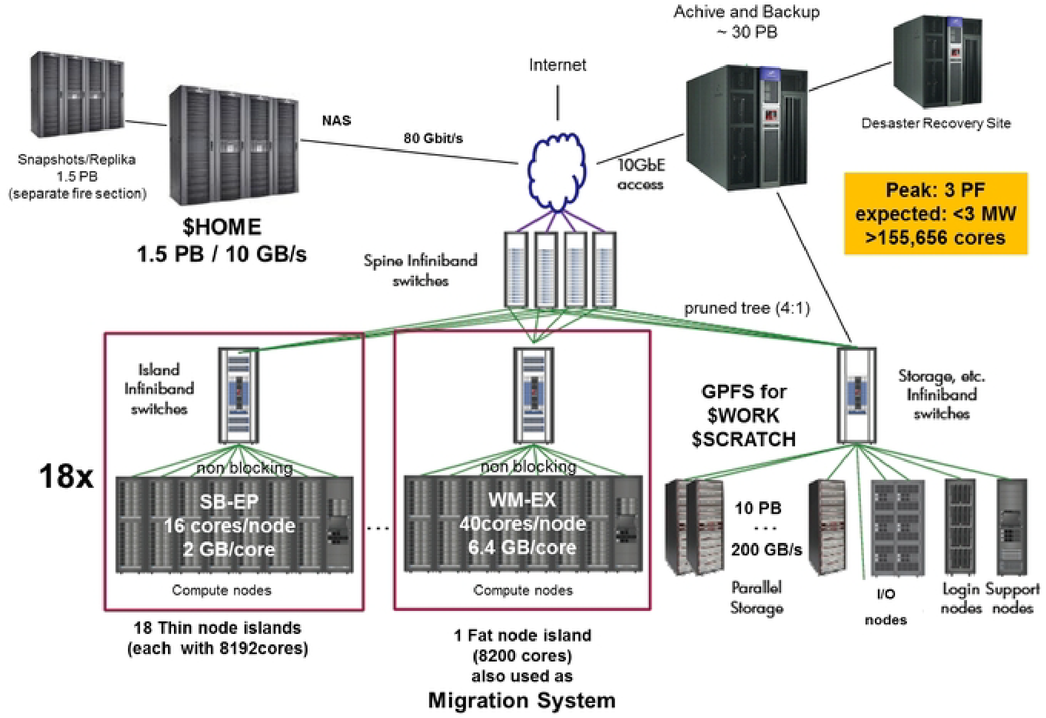

Figure 5. Schematic view of SuperMUC at LRZ (see www.lrz.de/services/compute/supermuc/systemdescription/, last accessed at 29th of May 2014)

instance GB/Watt. Hence, there are several different metrics that have to be considered for SuperMUC as a complete system.

Example for Gap 2 – In the SuperMUC scenario, changing the CPU type to achieve a higher energy efficiency would have (strong) side effects on other components of SuperMUC. For instance, using CPUs with a smaller L2 cache size might improve the CPU energy efficiency, but at the same time, SuperMUC's system interconnect between the CPUs and the non node-local memory will have higher workloads and therefore, its energy efficiency is decreased. This may lead to a decreased overall energy efficiency.

Example for Gap 3 – Bavaria in southern Germany, where SuperMUC is operated, is a relatively warm region compared to, for example, Iceland. The location therefore influences the demand for cooling and climate infrastructure. SuperMUC works energy-efficient because it supports hot-liquid cooling, i.e., free cooling from LRZ's rooftops can be used throughout the year without the demand for energy-demanding cooling machines. However, if SuperMUC was operated in Iceland, cooling it with fresh cold air from outside may be even more energy-efficient.

Example for Gap 4 – LRZ performs many different measurements regarding energy consumption and efficiency of both SuperMUC and the other IT infrastructure. However, decisions about further improvements have to be made manually by individuals from different departments as there is no common understanding of LRZ-wide EE yet and there is a complete lack of tool support when it comes to anything more than the simple visualization of raw measurement data.

It should also be mentioned that SuperMUC is one area in which EE is actively researched. EE for other parts of LRZ's IT infrastructure is hard to improve. For example, research papers have often suggested to turn network links between routers and switches off, e.g., outside office hours, to lower the energy consumption of the networking infrastructure. This does not work in practice for the simple fact that during the night often more traffic is generated than during the day, for example, due to automated backups and other bulk data transfers. Also, in an academic environment, it is impossible to completely shut down the networking infrastructure for whole building, e.g., over the weekend or holidays, because some researchers might still be working and depend on a working infrastructure.

2) Performance: Higher PE for the IT services that support business processes is the primary driver for investment in new and additional hardware and software. However, benchmarking and scaling PE often is tricky. For example, a computationally intensive application may benefit from faster CPUs and additional RAM, whereas a database server may best be sped up by replacing HDDs with SDDs; also, increasing the LAN bandwidth from 1 Gbit/s to 10 Gbit/s does not imply that employees have ten time faster access to local file servers or Internet content.

At LRZ and in the Munich Scientific Network, performance is critical for user experience: Students and faculty expect, for example, access to LRZ's central file servers from labs and offices to be as fast as locally operated storage solutions. However, the central file servers need to accommodate many more users who are active in parallel and the communication network needs to transport all the data across the backbone and access networks with the same quality-of-service parameters as a LAN, for example, regarding bandwidth and IP packet delay.

3) Information security: The primary goal of *Information Security* (IS) is to ensure the confidentiality, integrity, and availability of services and data. For example, the highly innovative research carried out on SuperMUC must not leak to unauthorized third parties and an attacker must be prevented from manipulating code, input as well as output data of HPC job submissions. Consequently, IS is an essential area of responsibility for both system administrators and management because of two reasons. First, it is a key component for compliance, i.e., the fulfillment of laws, like Germany's strict data protection and privacy laws, industry-sector-specific regulations, contracts with business partners, and intra-organizational policies. Second, many university departments and chairs store project data in cooperation with industry partners, resulting in high confidentiality, integrity, and availability demands. Measuring IS and providing adequate evidence even to third parties becomes more and more important. Despite this important role, IS often is perceived as a necessary evil, especially from the management perspective, because it costs money but, unlike other investment, cannot generate any direct return on invest (ROI) due to its nature. The aspects that are in the focus of IS are inherently hard to quantify because there are no standardized units of measurement yet. While many security experts have a reliable gut feeling about the security state of a system they analyze and there are many standardized IS controls, e.g., those specified in ISO/IEC 27001 [42], objectively assessing arbitrary security properties and making them comparable across organizations' boundaries is still impractical.

Concerning IS metrics, LRZ uses more than 50 measurement procedures and metrics to monitor the overall security level of its infrastructure. For example, regarding system management the delay between the vendor publication of software security patches/updates and their application to at least 80 percent of all relevant LRZ servers is measured. Each server's network traffic is monitored for suspicious IP packets and changes to its communication characteristics, which may indicate a compromised machine. *Virtual Private Network* (VPN) and *Wireless* (WiFi) users are monitored for Internet SMTP connections, and if certain thresholds are exceeded, these client machines are flagged as probably malware-infected, Spam-sending devices and are put into quarantine.

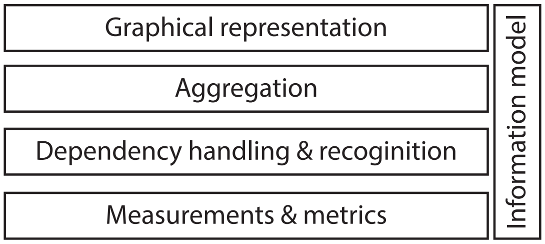

Figure 6. The presented design concept applies the layer pattern to achieve a separation of concern

4) Difficulties with metrics handling in practice: The results of each of the metrics categories discussed above are currently assembled and evaluated by different personnel: An EE project team works closely together with facility management, PE is handled by system administrators together with a central server operations group, and an inter-department security team handles IS and risk management.

For comprehensible pragmatic reasons, for example the security team does not always pay attention to EE issues, and software and hardware changes that are supposed to enhance PE do not always keep IS in mind. For design decisions on all layers of the organizational hierarchy, a holistic metrics management approach therefore would lead to more profound and even better results.

IV. A design concept for an integrated management platform of IT infrastructure metrics

This section presents a design concept for an integrated management platform of IT infrastructure metrics. The design concept aims at implementing the outlined design criteria (cf. Section II) in general and at closing the identified and details four gaps in particular.

Figure 6 depicts the design concept's four layers. As the layer titles imply, each layer has a dedicated topic. The undermost layer covers existing measurements and metrics and their integration. The next layer addresses the handling, use, and recognition of dependencies between the elements of the undermost layer. Based on that the third layer defines and implements aggregation logic whose results are (graphically) represented by the uppermost layer. Orthogonal to the four layers, the information model describes all relevant entities and their attributes.

The layer structure depicted in Figure 6 also guides the section's structure: We discuss the measurements and metrics layer in Section IV-A, followed by a description of a middle layer that handles dependencies detection and management in Section IV-B. The logic for aggregating and combining multiple measurements and metrics is then described in Section IV-C. Finally, Sections IV-D and IV-E deal with various aspects of the graphical representation of the results, which constitute the top layer of the architecture, and presents our information model in depth.

A. Layer 1 – Measurements and metrics

The undermost layer 1 comprises the measurements and metrics that provide the data, which is processed by all upper layers. Hence, errors and discrepancies in the initial "raw" data might raise to higher power, depending on the applied aggregation and processing algorithms. Consequently, a data quality as high as possible should be achieved. Layer 1 implements this role by considering *measurement quality* and *metric quality*.

Measurement quality – The most important aspect about measurements is the accuracy of the compiled values. This accuracy can be penalized by technical and social factors. Technical factors cover the measurement setup, e.g., the accuracy of the applied instruments or errors while storing the values. These problems can be addressed by a thoroughly planned measurement setup. Additionally, for each instrument or measurement procedure, the accuracy should be provided. For instance, most power consumption instruments like multimeters provide an accuracy about 2%. More challenging is the avoidance of social factors, especially avoiding the *feedback mechanism*: since measurement and metric results might affect the measuring entity in a negative way, the entity might (subconsciously) influence the measurement [19]. In other words, a measurement should be stable, i.e., compile the same results even if different entities or persons conduct the measurement [20].

Metric quality – The definition and enforcement of metric quality is a broad field and there are manifold approaches. For layer 1, we select the subsequently itemized definitions:

SMART A metric is considered to be good if it is *specific*, *measurable*, *attainable*, *repeatable*, and *time-Dependent* (SMART) [13], [43], [44, 6–10]. This set of quality criteria is in close correlation to the criteria defined by Bianzino et al. [24], i.e., *simple* (enough to understand), *accurate* (enough to withstand scrutiny), *usable* and *relevant* (enough to be an effective agent of change).

Stability A metric's semantic must remain the same during the entire life cycle and/or use time [45]. Additionally, the semantics is independent from the concrete description language, like QML, Windows Management Instrumentation or vendor specific SLA management solutions.

Empirical validation A metric must be defined in a way that it can be validated efficiently and empirically [45]. General speaking, a metric is of no use if it is not possible to validate its implementation or application [45]. For instance, defining "response time" as metric also requires a possibility to check the response time empirically.

B. Layer 2 – Dependency handling and recognition

The layer 2 covers the topic of handling and recognizing dependencies between two or multiple metrics. Dependencies are split in *reciprocity*–dependencies and *aggregation*–dependencies: former describes correlations between metrics, for instance, improving CPU energy efficiency potentially decreases interconnect energy efficiency. The latter addresses the aggregation of metrics to form new statements. Both dependency types comprise a *definition* phase and a *detection* phase.

The **definition** phase of a dependency covers the semantics of a dependency and influences the detection and modeling. Basically, there are different types of dependencies according to the considered attribute categories and hardware types. Additionally, there are direct and indirect dependencies. The direct dependencies affect the metric itself, for instance, the current load of a hardware component influences the measurements and metrics about time to completion or current power consumption. Indirect dependencies are between the hardware components and hence, affect the applied measurements and metrics only indirectly.

According to the definition, there are different ways of dependency **detection**, i.e., analytically or empirically. An analytic detection mechanism processes (structural) information about the considered IT infrastructure, like a *Configuration Management Database* (CMDB) [46], and derives dependency insights. An empirical detection mechanism collects data at different points in time at different sensor points in the IT infrastructure, e.g., before and after a reconfiguration. Another example is the mechanism of failure injection as applied by Bagchi et al. for uncovering resource dependencies in a dynamic distributed e-commerce environment [47].

C. Layer 3 – Aggregation logic

Layer 3 uses the information provided by the two layers below, i.e., the (revised) raw measurement and metrics data provided by layer 1 and the (incorporated) dependency information generated by layer 2. The definition and implementation of aggregation rules and concepts are encapsulated in layer 3 and split in three aspects, i.e., the aggregation *direction*, the applied aggregation *rules* and the aggregation rule *declaration*.

There are three possible aggregation **directions**, i.e., bottom-up, hypothesis generation on middle, and top-down. *Bottom-up* aggregation uses existing data from low abstraction levels and aggregate them iteratively until the pursued granularity level is reached. The most difficult task while doing a bottom-up generation is the "correct" selection of attributes/values at the lowest level. *Hypothesis generation* formulates hypotheses on an intermediate level and tries to prove or disprove those hypotheses by applying data from low abstraction levels. Those (dis)proved hypotheses are afterwards used to generate statements for a higher-level consideration. *Top-down* starts at certain points in the upper levels and tries to create the data tree beginning at the root by recursively finding suitable metrics on the next lower

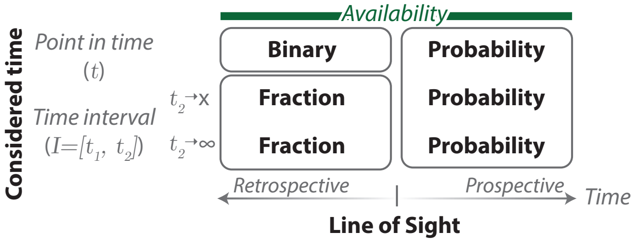

Figure 7. Metrics about component availability result in *binary*, *fraction*, and *probability* data types (taken from [49])

level. The aggregation logic layer applies the *Bottom-up* direction, since we aim at integrating existing measurements and metrics and hence, have to start at this (immutable) point.

Having set the aggregation direction, the next step is the definition of aggregation **rules** that describe metric aggregation (in a mathematical way). As explained in Section II-A, a metric is a mapping of two measurement values on a distance, which is the *image* of that mapping. To cover the variety of existing measurements and metrics as well as the different metric categories (cf. Section I), aggregation focuses solely on that *image*. This narrows the problem domain of aggregation to a set of very basic data types and semantics. e.g., *binary*, *fraction*, and *probability*.

Figure 7 explains these data types exemplary for the metric category *availability* that describes a component being in the ***up state***, i.e., it delivers a *correct service* or a *system function* as it is described by the functional specification [48]. The line of sight describes whether the considered availability values are in the past or in the future. The considered time describes whether a component's availability is considered for a discrete point in time or a time period. Bringing these dimensions together, only *binary*, *fraction*, and *probability* values are possible. For instance, a component was (*retrospective*) available at t (*point in time*) yes or no (*binary*). According to the data type and the implicitly contained semantics the aggregation rules can be formulated. An extended explanation and further details are provided in [49].

Finally, the selected aggregation rules must be **declared**. Basically, an arbitrary language for rule declaration can be applied, as long as it meets the following requirements that were developed in previous work in our research group by Sailer (see [50]):

1) Expressiveness: A declarative programming language shall be used; this enhances the legibility of the metrics specification, e.g., compared to XML-based specifications, and is sufficiently decoupled from specific implementations.
2) Access to data: Any derived measure or metric is a synthesis of data retrieved from various sources. The used language must make this data available, e.g., as read-only variables.
3) Aggregation operations: Many metrics can be expressed using basic arithmetic operations. However, more complex metrics require statistical function libraries and first-order logic. Ideally, language users can define their own functions.
4) Triggers: To ensure that accessed data is up-to-date and eventually trigger other preparations of the environment before aggregation operations are performed, the interaction capabilities of the used language must include ways to start and control measurements and other processes.

We propose to use the *Service Information Specification Language* (SISL) that has been introduced by Danciu et al. [51]. It has explicitly been designed independent of specific IT systems, metrics categories, or implementation technologies. It is strictly typed and provides support for integers, floating point numbers, strings, and temporal as well as Boolean expressions.

D. Layer 4 – Graphical representation

According to one of the fundamental design principles of software development, i.e., separating (graphical) representation and logic, layer 4 encapsulates the graphical representation of the results compiled by the other three layers. The implementation of the layer 4 highly depends on the individual objectives of top-level management, the required insights for decision making and the characteristics of the information provided by layer 3.

Figure 8 depicts an exemplary graphical representation of information about the energy efficiency of LRZ's SuperMUC (cf. Section III). The depicted management cockpit comprises three areas, i.e., a *tree-view* for aggregated values (labeled "1"), a *delta-view* of current and oblige values (labeled "2"), and a *activity recommendation* (labeled "3").

The **tree-view** provides information about the sources of a particular value. This information is required by the urgent need of provenance and to facilitate root cause analysis: starting at the top level, any aggregated metrics value can be broken down into smaller pieces and it can be explained how this high-level current value materializes. Figure 8 illustrates that the overall energy efficiency value of SuperMUC is aggregated from *Interconnect*, *Storage*, and *CPU* values. The CPU value, in turn, is composed of *Operations/kWh* and *MFlops/Watt* values.

The **delta-view** compares the current (aggregated) value and its assigned obliged value. The delta's color is determined by the predefined allowed threshold for a particular metric or rather its interpretation: if the threshold is exceeded, the delta is colored red. For each perspective (cf. Section II-C) a different set oblige values can be defined.

The **activity recommendations** depend on the delta of oblige and current values, a optionally predefined escalation mechanism or a criticality level. A possible recommendation could be to decrease the CPU clock time. Obviously, the

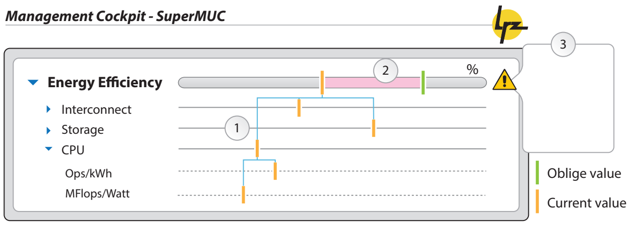

Figure 8. Exemplary graphical representation of high-level, management-relevant information about SuperMUC's energy efficiency.

challenges described in the activity gap (cf. Section II-B) have to be thoroughly incorporated.

E. Information model

As further detailed in the following Section V, the strictly separated development of metrics in different research areas and the absence of inter-domain metrics aggregation concepts results in a lack of existing common information models, standards, and best practices for the internal representation of measures, aggregation rules, reports, and other ways to refine and present metrics to decision makers and other users. Hence, a uniform information (and data) model for the various metrics categories is imperative for scalability.

This uniform information model is provided by the orthogonal layer *Information model* that covers all important elements and attributes of the above detailed four layers. Depending on the implementation technology, a platform specific data model can easily be derived from the provided information model. Figure 9 depicts the information model as *Unified Modeling Language* (UML) class diagram. Guided by the overall article's aim of providing a design concept but concrete implementations (cf. Section I), the class diagram representation is chosen to ease extending the provided information model by individual sub classes. For the same reason, some classes come with less attributes, like a Role, since the set of reasonable attributes depends on the individual situation.

The classes are organized in three packages, i.e., general, datasources, and representation, which are subsequently further detailed.

general package – The package contains all classes that are relevant for storing meta information about the core classes. The class UniqueID provides a globally unique identifier to ensure an ambiguous identification of each particular element. Further description of entities is achieved by assigning an arbitrary set of Keyword and Category objects. The classes Timestamp, Formula and Frequency provide (complex) data types and describe their representation. A Formula, for instance, has a natural language label and description and a specific way of declaration it (cf. Section IV-C). A Role is used to describe a responsibility, e.g., a system administrator or laboratory employee.

datasource package – This is the main package, since it contains all elements for gaining, gathering and compiling information for the management cockpit. The contained classes are further structured in measurement, metric and interpretation, guided by the concepts presented in Figure 3 (page 4).

The Datasource class collects all attributes that are in common for a measurement, a metric, and an interpretation. Plain management aspects are described by a label and the objective in natural language and an arbitrary number of Keyword and Category objects to facilitate searches for suitable measurements and metrics, e.g., if new reports have to be designed. Additionally, a version is stored to allow the application of different versions at the same time and to support provenance. The version information is enhanced by a DatasourceStatus enumeration, comprising items such as Active or Retired.

Responsibilities are described by an arbitrary set of Role objects and the accordant association class. In this Responsibility class, the responsibility can be detailed, e.g., performing a measurement, reviewing a metric or being the authoritative source for a measurement or metric, like a SLA or policy.

To enable a reuse of scales, there is a dedicated class Scale. Besides a natural language label and description, the Scale class most importantly describes a unit. Exemplary values are *Watt* (for a measurement), *Ops/kWh* (for a metric) or *school grade* (for an interpretation). To further detail the scale, a ScaleType enumeration entry can be assigned.

Besides the above detailed general elements, the datasource package contains additional packages for each element depicted in Figure 3, i.e., a measurement package, a metric package, and an interpretation package.

measurement package – All entities relevant for measurement and storing measurement values are collected in this package. The Measurement class describes the activity of measuring or in other words, the mapping of facts to a symbol set (cf. Figure 3). Consequently, the class contains information about the measurement activity, i.e., what (measuredComponent) was when (timestamp) how (isAutomated) measured. The applied procedure is further detailed by the MeasurementProcedure class. The frequency of reviewing the measurement activity, e.g., analyzing the applied procedure or re-checking for necessity and suitability, is described in the reviewFrequency.

The compiled measurement values or the image of the mapping (cf. Figure 3) are stored in MeasurementValue objects. The class' value attribute is dependent on the assigned scale. The differentiation between simple and derived measurement values introduced in Section II-A is represented by the dedicated class DerivedMeasurementValue that is assigned to an

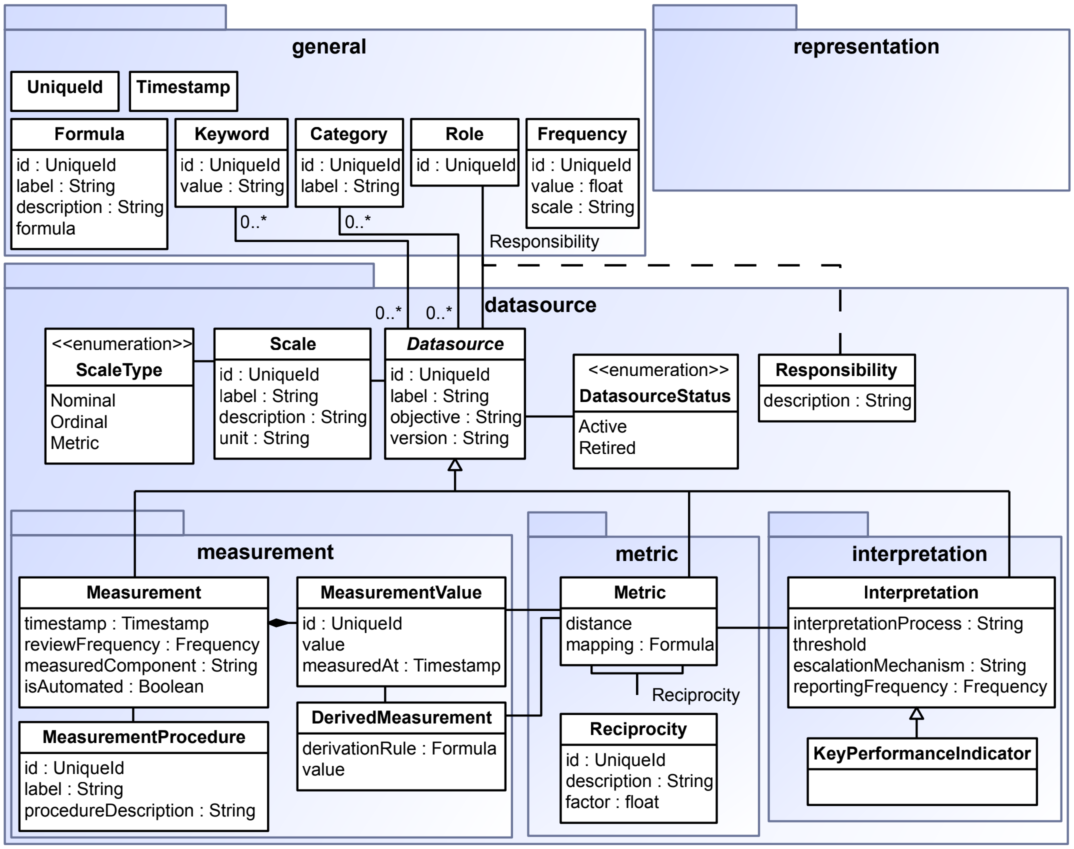

Figure 9. The design concept's information model.

arbitrary set of MeasurementValues and describes a derivationRule.

metric package – The package contains elements to describe the mapping of two measurement values on a distance and related aspects (cf. Figure 3). The Metric class stores a mapping of the assigned measurement values to the compiled distance. Dependencies between metrics are modeled by the Reciprocity, which contains a natural language description and a factor defining how two metrics are related. A factor value of 2, for instance, would describe a positive correlation by the factor 2.

interpretation package – The elements to describe interpretations for a metric (cf. Section II-A) are summarized in the interpretation package. The Interpretation describes a interpretationProcess, i.e., how the interpretation is conducted. To support automated interpretation, it additionally contains a threshold and an escalationMechanism, which is triggered on threshold violations. The escalationMechanism is not contained in the Datasource class but only assigned to the Interpretation, since we are interested in top-level management decision support and not in low-level monitoring, which mostly already applies mature escalation and trigger mechanisms. A reportingFrequency can be specified if the interpretation is not only used interactively via the management cockpit, but also included in periodic reports.

The Interpretation is refined by a KeyPerformanceIndicator class, which is typically required for organization-internal audits or are included in reports. This class could be used as interface to ITSM processes, which can be based, for example, on the *IT Infrastructure Library* (ITIL v3 [52]) or the ISO/IEC 20000-1 standard [53]: IT service providers have contracts with their customers, which are referred to as SLAs, and each SLA typically specifies thresholds and nominal values for KPIs, e.g., the service's monthly availability must at least be 99.9 percent. SLA violations by the service provider can then, for example, lead to penalties.

representation package – The above mentioned software design principle of separating logic and representation guided not only the layered architecture of the presented design concept, but also the package structure depicted in Figure 9. As described in layer 4 (cf. Section IV-D) the graphical representation heavily depends on several individual parameters. Consequently, the representation package is only a placeholder to emphasize the urgent need to separate information representation from the other elements. Hence, the package contains all components that are necessary to achieve the graphical representation. For the example provided in Figure 8, there could be color assignments or labels. Further examples are suggestions for *graphic representation* could be stored, e.g., whether the measure's history should be visualized as line chart, *interpretation rules*, i.e., guidelines for understanding what the measure expresses, *decision rules*, i.e., guidelines for acting appropriately based on changes to the measure or a lack thereof, and *instructions*, i.e., suggestions for actions that should be taken depending on which decisions have been made.

V. Metrics management in related work

In the following subsections, we discuss related work and the current state-of-the-art. Discussion covers a quantitative additive metric, i.e., *energy efficiency*, and a qualitative metric, i.e., *information security*. Additionally, approaches for structuring metrics and aggregating their values are considered.

A. Energy efficiency metrics

There are a lot of metrics dealing with different energy efficiency aspects, e.g., measuring the power consumption of computing servers and clusters [54], [55], or the power consumption in optical IP networks [56]. Further examples are TEEER [57], EPI [58], ECR, and ECRW [59], [60], [61]. All of them are defined by providing calculation and interpretation rules, partially in a very comprehensive way, but nevertheless they all focus on technical aspects of a single entity on a very low level. Hence, they do not facilitate a holistic view on the energy efficiency situation of SuperMUC, which, being a large HPC system, aggregates many different hardware components in a complex architecture. There is some work postulating the extension of existing metrics, e.g., Banerjee et al. [58] propose to define energy consumption not in an absolute way, but proportional to the workload.

B. Security metrics

The work on security metrics, which earned its spot on the INFOSEC research council's *hard problems list* in 2005 [62], is motivated by the difficulty of answering seemingly simple questions, such as: Which of the n possible system configuration variations is the most secure? Is it advisable to invest in security measure x? Is organization i's security level higher than organization j's? As there is no physical unit of measurement for security and we still lack established standards and best practices, the currently only commonly accepted conclusion is that each single security measurement or security metrics has limited expressiveness [63].

The most wide-spread approach to use IS metrics is hypothesis-based [64], [65]: Hypotheses are derived from known risks or attacker models, and metrics are defined to corroborate or vitiate them. Many dozens of security metrics have been suggested, e.g., by [64] and [66]. A common denominator of many IS metrics is that the involved units are currencies or durations; this facilitates a direct mapping to operational costs or amount of work, which often is preferred by top management according to [64]. NIST has published its *directions in security metrics research* in 2009 [67] and defined milestones for the improvement of security measurability.

IS metrics are closely related to investment models, such as Gordon's and Loeb's [68], which allows for ex-ante security measure cost-benefit calculations. It is motivated by the problem that classic economic models, which typically involve some sort of return on invest (ROI), are unsuitable for IS investments because security measures usually cannot directly increase the volume of sales or profit; instead, they only impede or reduce the effects of security events causing damage.

Security is, especially due to the heterogeneity of existing metrics, probably not the youngest research area for measurements and metrics, but the most complex one of the three we investigate. This assumption is supported by the almost complete lack of IS metrics management software so far. Several researchers and commercial vendors have attempted to adapt existing security management software, such as security information & event management (SIEM) systems, for security metrics and security report creation purposes.

However, as already discussed by Jaquith in [64], such systems have a strong focus on real-time monitoring, whereas security metrics are intended to facilitate long-term processes and decision making. They also focus on single measurements or the extraction of information from log entries, whereas several security measurements typically need to be aggregated and combined to form a security metric. IS metrics are not necessarily purely technical either, for example when the percentage of employees who already received the quarterly security instructions is calculated, whereas SIEM systems and similar solutions focus on technical measurements only. A comprehensive tool set for integrated metrics management therefore would highly benefit IS.

C. Aggregation of measurements and metrics

According to the basic ideas presented for layer 3 in the previous section, the most important aspect for aggregating metrics is the metric image. Consequently, some structuring and taxonomy approaches are considered, since their results could be used to gain insights in a particular metric's image.

There is literature and ongoing research in several topics about metrics taxonomy [69], [70], classification [24], and comparison [13]. These approaches structure and compare the aforementioned metrics in a single specific metric category or granularity level, like equipment-level metrics, and confine themselves to comparison. Therefore, they do not allow a holistic view either.

There are some comprehensive papers that compare and classify existing metrics, like [24], which proposes four hierarchical levels "equipment", "facility", "corporate", and "country". Wang et al. [70] recommends server level benchmarks and data centre benchmarks. Also these approaches focus on a specific class of metrics, like Bianzino et al. [24] do on equipment-level, and confine themselves to comparing single metrics. Therefore, they do not allow a holistic view neither.

VI. Summary and outlook

In this article, we first outlined the importance of accurate information about complex IT infrastructures to support profound decision making. Quantitatively expressing the key properties of IT systems, and aggregating raw measurements to meaningful higher-level information is a non-trivial task when data from various domains, such as energy efficiency, performance, and security have to be integrated. After introducing the basic terminology, we conducted a gap analysis with the following four key results: First, there is an information gap, i.e., information provided by most current metrics is not suitable for decision making on higher abstraction levels. Second, the interdependencies between the metrics are not sufficiently considered. Third, there currently is no support for the derivation of concrete improvement recommendations based on the metrics values. And fourth, existing approaches do not allow for customization in order to incorporate organization-specific requirements. To exemplify these four gaps, we outlined the Leibniz Supercomputing Centre scenario and described how measurements and metrics are applied in practice currently, along with the drawbacks that result from a non-integrated approach.

Motivated by these deficiencies, we then presented our design concept for an integrated management platform for IT infrastructure metrics. The overall design is based on a four-layered architecture, which we described in detail: On the lowest layer, 1, measurements and metrics are handled. Layer 2 then deals with the recognition and handling of dependencies between two or more metrics. Layer 3 uses the information provided by the lower layers to conduct the required aggregation logic, and Layer 4 covers important aspects of the graphical representation of the *management cockpit*. Our information model describes all relevant entities and their attributes as they are used across those four layers. Finally, we investigated the state of the art of metrics management for energy efficiency and security metrics as well as for metrics aggregation.

Our ongoing work will focus on the following open issues next:

Target values and comparison In order to provide "Warnings and activity recommendations", target values and interpretation rules for a delta between those target values and current values are mandatory. We have to investigate how to define or rather find those target values. This step is very critical, because having wrong target values would lead to optimizing the infrastructure towards wrong values. Additionally, we have to analyze how to interpret a delta between the current value and the target value for any given metric. This interpretation has three dimensions: overall meaning, timing aspects (e.g., "delta implies the necessity to act immediately", "delta is just for the annual, paper-based report"), and impact (e.g., "the severity of the delta is very high", "solving the delta is very costly").

Validation We need to perform a practical evaluation of our approach, i.e., the metrics in use today in our scenario need to be analyzed for their interdependencies and implemented based on our information model. This will serve as a basis for a prototype implementation of the management cockpit, which will be used to demonstrate the benefits of our solution in a real-world scenario. We will then also include metrics from additional categories and analyze the scalability of our integrated management approach.

Prospective view The management cockpit presented in this article uses measurement data, i.e., data about the (recent) past. Enabling comprehensive *what-if* analysis about planned modifications would require the application of models and their compiled predictions, i.e., data about the future. Consequently, we currently investigate the integration of existing models for manifold IT infrastructure types and architectures.

Acknowledgment

The authors thank the members of the Munich Network Management (MNM) Team (www.mnm-team.org) for valuable comments on previous versions of this article. The MNM-Team, directed by Prof. Dr. Dieter Kranzlmüller and Prof. Dr. Heinz-Gerd Hegering, is a group of researchers from the Ludwig-Maximilians-University Munich, the Technische Universität München, the Munich University of the German Federal Armed Forces, and the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities.

References

[1] C. Straube, W. Hommel, and D. Kranzlmüller, "A Platform for the Integrated Management of IT Infrastructure Metrics (Best Paper Award)," in *Proceedings of the 2nd International Conference on Advanced Communications and Computation (INFOCOMP'12)*, 2012, pp. 125–129.

[2] L. Xue, G. Ray, and B. Gu, "Environmental Uncertainty and IT Infrastructure Governance: A Curvilinear Relationship," *Information Systems Research (INFORMS)*, vol. 22, no. 2, pp. 389–399, 2011.

[3] S. H. Chung, T. A. Byrd, B. R. Lewis, and F. N. Ford, "An Empirical Study of the Relationships between IT Infrastructure Flexibility, Mass Customization, and Business Performance," *ACM Special Interest Group on Management Information Systems (SIGMIS)*, vol. 36, no. 3, pp. 26–44, 2005.

[4] P. Weill, "The Role and Value of Information Technology Infrastructure: Some Empirical Observations," Massachusetts Institute of Technology (MIT), Sloan School of Management, Tech. Rep. Sloan WP No. 3433-92, 1992.

[5] D. Briody, "Big Data Harnessing a Game-Changing Asset – A Report from the Economist Intelligence Unit Sponsored by SAS," Economist Intelligence Unit, Tech. Rep., 2011.

[6] C. Rupp, *Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis*. München: Hanser, 2009, vol. 5.

[7] S. Kleuker, *Grundkurs Software-Engineering mit UML: Der pragmatische Weg zu erfolgreichen Softwareprojekten*. Springer, 2013, vol. 3.

[8] I. Jacobson, *Object Oriented Software Engineering: A Use Case Driven Approach*. Wokingham, UK: Addison-Wesley, 1992.

[9] N. B. Duncan, "Capturing Flexibility of Information Technology Infrastructure: A Study of Resource Characteristics and their Measure," *Journal of Management Information Systems*, vol. 12, no. 2, pp. 37–57, 1995.

[10] T. A. Byrd and D. E. Turner, "Measuring the Flexibility of Information Technology Infrastructure: Exploratory Analysis of a Construct," *Journal of Management Information Systems*, vol. 17, no. 1, pp. 167–208, 2000.

[11] S. Laan, *IT Infrastructure Architecture – Infrastructure Building Blocks and Concepts*. Lulu Press, Inc., 2011.

[12] H. Schackmann and H. Lichter, "Process Assessment by Evaluating Configuration and Change Request Management Systems," in *Proceedings of the Warm Up Workshop for ACM/IEEE ICSE 2010 (WUP'09)*, 2009.

[13] S. C. Payne, "A Guide to Security Metrics," SANS Institute, Tech. Rep., 2006.

[14] M. Broadbent and P. Weill, "Management by Maxim: How Business and IT Managers Can Create IT Infrastructures," *Sloan Management Review*, vol. 38, no. 3, pp. 77–92, 1997.

[15] C. Soanes and A. Stevenson, *Oxford Dictionary of English*. Oxford University Press, 2010, vol. 3.

[16] P. Weill, M. Subramani, and M. Broadbent, "IT Infrastructure for Strategic Agility," Massachusetts Institute of Technology (MIT), Sloan School of Management, Tech. Rep., 2002.

[17] C. Straube and D. Kranzlmüller, "An Approach for System Workload Calculation," in *Proceedings of the 12th International IASTED Conference on Parallel and Distributed Computing and Networks (PDCN'14)*, 2014.

[18] R. Böhme and F. C. Freiling, "On Metrics and Measurements," in *Dependability Metrics*, I. Eusgeld, F. C. Freiling, and R. Reussner, Eds. Springer, November 2008, pp. 7–13.

[19] R. Böhme and R. Reussner, "Validation of Predictions with Measurements," in *Dependability Metrics*, I. Eusgeld, F. C. Freiling, and R. Reussner, Eds. Springer, November 2008, pp. 14–18.

[20] H. Koziolek, "Goal, Question, Metric," in *Dependability Metrics*, I. Eusgeld, F. C. Freiling, and R. Reussner, Eds. Springer, November 2008, pp. 39–42.

[21] C. Villarrubia, E. Fernández-Medina, and M. Piattini, "Metrics of Password Management Policy," in *Conference on Computational Science and Its Applications (ICCSA'06)*. Springer, 2006, vol. 3982.

[22] R. Ramler and K. Wolfmaier, "Issues and Effort in Integrating Data from Heterogeneous Software Repositories and Corporate Databases," in *Proceedings of the 2nd International ACM/IEEE Symposium on Empirical Software Engineering and Measurement (ESEM'08)*, 2008.

[23] C. Schaller, A. C. Neuroni, D. Mares, R. Riedl, and S. Urs, "Cockpits for Swiss Municipalities: a Web Based Instrument for Leadership," in *Proceedings of the 11th International ACM Digital Government Research Conference on Public Administration Online: Challenges and Opportunities (dg.o'10)*, 2010.

[24] Aruna Prem Bianzino and Anand Kishore Raju and Dario Rossi, "Apples-to-Apples: a Framework Analysis for Energy-Efficiency in Networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 3, pp. 81–85, 2010.

[25] D. Chen, N. Eisley, P. Heidelberger, S. Kumar, A. Mamidala, F. Petrini, R. Senger, Y. Sugawara, R. Walkup, B. Steinmacher-Burow, A. Choudhury, Y. Sabharwal, S. Singhal, and J. Parker, "Looking Under the Hood of the IBM Blue Gene/Q Network," in *Proceedings of the International ACM/IEEE Conference on High Performance Computing, Networking, Storage and Analysis (SC'12)*, 2012.

[26] T. Hoefler, T. Mehlan, A. Lumsdaine, and W. Rehm, "Netgauge: A Network Performance Measurement Framework," in *Proceedings of the High Performance Computing and Communications (HPCC'07)*, L. Yang, R. Mello, J. Subhlok, B. Chapman, and R. Perrott, Eds. Springer, 2007, vol. 4782, pp. 659–671.

[27] M. Meswani, M. Laurenzano, L. Carrington, and A. Snavely, "Modeling and Predicting Disk I/O Time of HPC Applications," in *Proceedings of the High Performance Computing Modernization Program Users Group Conference (HPCMP-UGC)*, 2010.

[28] J. Elliott, K. Kharbas, D. Fiala, F. Mueller, K. Ferreira, and C. Engelmann, "Combining Partial Redundancy and Checkpointing for HPC," in *Proceedings of the 32th International IEEE Conference on Distributed Computing Systems (ICDCS)*, 2012.

[29] I. Eusgeld, B. Fechner, F. Salfner, M. Walter, P. Limbourg, and L. Zhang, "Hardware Reliability," in *Dependability Metrics*, R. Reussner, F. C. Freiling, and I. Eusgeld, Eds. Springer, 2008, vol. 4909, pp. 59–103.

[30] B. Farbey, D. Targett, and F. Land, “The Great IT Benefit Hunt,” *European Management Journal*, vol. 12, no. 3, pp. 270–279, 1994.

[31] M. Al-Mashari and M. Zairi, “Creating a Fit Between BPR and IT Infrastructure: A Proposed Framework for Effective Implementation,” *Journal of Flexible Manufacturing Systems*, vol. 12, no. 4, pp. 253–274, 2000.

[32] C. Straube and D. Kranzlmüller, “An IT-Infrastructure Capability Model,” in *Proceedings of the 10th ACM Conference on Computing Frontiers (CF’13)*. ACM, 2013.

[33] W. Smith, D. Acay, R. Fano, and G. Ratner, “Tools for Designing and Delivering Multiple-Perspective Scenarios,” in *Proceedings of the 18th ACM Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments (OZCHI’06)*, 2006.

[34] Klaus R. Dittrich and Patrick Ziegler, “Three Decades of Data Integration - All Problems Solved?” in *Proceedings of the 18th World Computer Congress on Building the Information Society (IFIP)*, 2004.

[35] David M Pennock and Alexandrin Popescul and Andrew I. Schein and Lyle H. Ungar, “Methods and Metrics for Cold-Start Recommendations,” in *Proceedings of the 25th International ACM Conference on Research and Development in Information Retrieval (SIGIR’02)*, 2002.

[36] H. Bauke and S. Mertens, *Cluster Computing: Praktische Einführung in das Hochleistungsrechnen Auf Linux-Clustern*. Springer, 2006, vol. 1.

[37] G. Contreras and M. Martonosi, “Power Prediction for Intel XScale Processors Using Performance Monitoring Unit Events,” in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED’05)*, 2005.

[38] X. Fan, W.-D. Weber, and L. A. Barroso, “Power Provisioning for a Warehouse-Sized Computer,” in *Proceedings of the 34th International Symposium on Computer Architecture (ISCA’07)*, 2007.

[39] M. Guest, *The Scientific Case for High Performance Computing in Europe 2012-2020*. Insight Publishers Ltd, 2013.

[40] D. Jensen and A. Rodrigues, “Embedded Systems and Exascale Computing,” *Computing in Science Engineering*, vol. 12, no. 6, pp. 20–29, 2010.

[41] D. Hitchcock and L. Nowell, “Advanced Architectures and Critical Technologies for Exascale Computing,” U.S. Department of Energy (DoE), Tech. Rep. DE-FOA-0000255, 2010.

[42] ISO/IEC 27001:2005, *Information technology – Security techniques – Information security management systems – Requirements*. International Organization for Standardization and International Electrotechnical Commission, 2005.

[43] “System Security Engineering - Capability Maturity Model – Model Description Document,” Carnegie Mellon University, Tech. Rep. 3.0, 2003.

[44] Reijo Savola, “Towards a Security Metrics Taxonomy for the Information and Communication Technology Industry,” in *Proceedings of the International IEEE Conference on Software Engineering Advances (ICSEA’07)*, 2007.

[45] Kenneth Chan and Iman Poernomo, “Consistent Metric Usage: From Design to Deployment,” in *Proceedings of the Dependability Metrics: Advanced Lectures [result from a Dagstuhl seminar]*, 2008.

[46] S. Knittl, “Werkzeugunterstützung für interorganisationales IT-Service-Management - ein Referenzmodell für die Erstellung einer ioCMDB,” Ph.D. dissertation, Technische Universität München (TUM), 2012.

[47] S. Bagchi, G. Kar, and J. Hellerstein, “Dependency Analysis in Distributed Systems using Fault Injection: Application to Problem Determination in an e-commerce Environment,” in *Proceedings of the 12th International Workshop on Distributed Systems (DSOM’01)*, A. Pras and O. Festor, Eds., 2001.

[48] A. Avizienis, J.-C. Laprie, and B. Randell, “Fundamental Concepts of Dependability,” University of California, Los Angeles (UCLA), Tech. Rep., 2001.

[49] C. Straube and D. Kranzlmüller, “Model-Driven Resilience Assessment of Modifications to HPC Infrastructures,” in *Proceedings of the 6th Workshop on Resiliency in High Performance Computing (Resilience) in Clusters, Clouds, and Grids in Conjunction with Euro-Par 2013*, 2013.

[50] M. Sailer, “Konzeption einer Service-MIB - Analyse und Spezifikation dienstorientierter Managementinformation,” Ph.D. dissertation, Ludwig-Maximilians-Universität München, 2007.

[51] V. A. Danciu, N. gentschen Felde, and M. Sailer, “Declarative Specification of Service Management Attributes,” in *Proceedings of the 10th International IFIP/IEEE Symposium on Integrated Network Management (IM’07)*, 2007.

[52] Office of Government Commerce (OGC), *IT Infrastructure Library v3: Service Design, 2nd impression*. ISBN 978-0113310470, The Stationery Office (TSO), 2007.

[53] ISO/IEC 20000-1:2005, *Information technology – Service management – Part 1: Specification*. International Organization for Standardization and International Electrotechnical Commission, 2005.

[54] Christos Kozyrakis and Parthasarathy Ranganathan and Suzanne Rivoire and Mehul A. Shah, “JouleSort: a Balanced Energy-Efficiency Benchmark,” in *Proceedings of the International ACM Conference on Management of Data (SIGMOD ’07)*, 2007.

[55] Christos Kozyrakis and Justin Meza and Parthasarathy Ranganathan and Suzanne Rivoire and Mehul A. Shah, “Models and Metrics to Enable Energy-Efficiency Optimizations,” *Computer*, vol. 40, no. 12, pp. 39–48, 2007.

[56] Robert Ayre and Jayant Baliga and Kerry Hinton and Wayne V. Sorin and Rodney S. Tucker, “Energy Consumption in Optical IP Networks,” *Lightwave Technology*, vol. 27, no. 13, pp. 2391–2403, 2009.

[57] T. Talbot and L. C. Graff, “Verizon NEBS TM Compliance: TEEER Metric Quantification,” Verizon Communications Inc., Tech. Rep. VZ.TPR.9207, 2009.

[58] Sujata Banerjee and Priya Mahadevan and Parthasarathy Ranganathan and Puneet Sharma, “A Power Benchmarking Framework for Network Devices,” in *Proceedings of the 8th International IFIP-TC 6 Networking Conference (NETWORKING ’09)*, 2009.

[59] Luc Ceuppens and Daniel Kharitonov and Alan Sardella, “Power Saving Strategies and Technologies in Network Equipment Opportunities and Challenges, Risk and Rewards,” in *Proceedings of the International IEEE Symposium on Applications and the Internet (SAINT’08)*, 2008.

[60] A. Alimian, B. Nordman, and D. Kharitonov, “Network and Telecom Equipment - Energy and Performance Assessment – Test Procedure and Measurement Methodology,” IXIA Corp / Lawrence Berkeley National Lab / Juniper Networks, Inc., Tech. Rep. Draft 1.0.4, 2008.

[61] “Energy Efficiency for Network Equipment: Two Steps beyond Greenwashing,” Juniper Networks, Inc., Tech. Rep., 2010.

[62] INFOSEC Research Council, “INFOSEC Hard Problem List,” http://www.infosec-research.org/docs_public/20051130-IRC-HPL-FINAL.pdf, 2005, [Online; accessed 30-May-2014].

[63] R. Böhme, “Security Metrics and Security Investment Models,” in *Proceedings of IWSEC 201, LNCS 6434*. Springer, 2010, pp. 10–24.

[64] Andrew Jaquith, *Security Metrics — Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley Longman, Amsterdam, ISBN 978-0321349989, 2007.

[65] H. Langweg, “Framework for Malware Resistance Metrics,” in *Proceedings of the 2nd Workshop on Quality of Protection (QoP’06)*. ACM, 2006.

[66] V. Ertürk, “A Framework Based on Continuous Security Monitoring,” Master Thesis, The Middle East Technical University, 2008.

[67] W. Jansen, “Directions in Security Metrics Research, NISTIR 7564,” National Institute of Standards and Technology Report, 2009.

[68] L. A. Gordon and M. P. Loeb, “The Economics of Information Security Investment,” *ACM Transactions on Information and System Security*, vol. 5, no. 4, pp. 438–457, 2002.

[69] Ronda Henning and Ambareen Siraj and Rayford B. Vaughn, Jr., “Information Assurance Measures and Metrics - State of Practice and Proposed Taxonomy,” in *Proceedings of the 36th International IEEE Hawaii Conference on System Sciences (HICSS’03)*, 2003.

[70] L. Wang and S. U. Khan, “Review of Performance Metrics for Green Data Centers: a Taxonomy Study,” *The Journal of Supercomputing*, vol. 63, no. 3, pp. 639–656, 2011.