

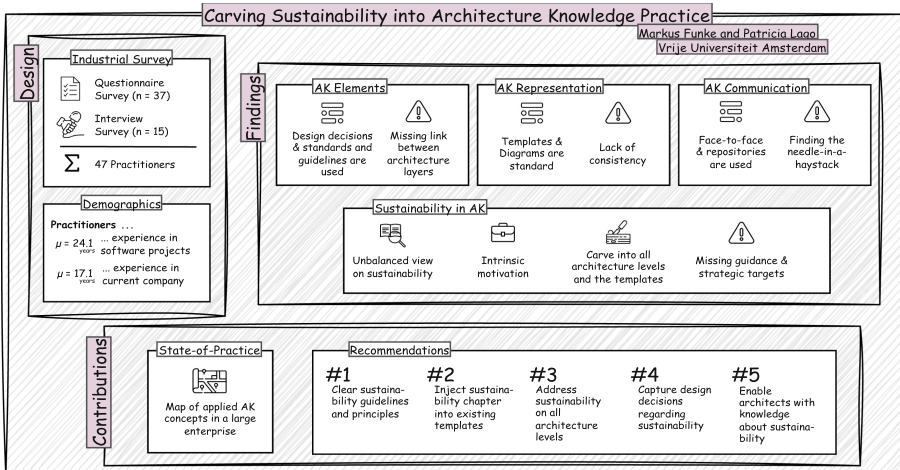


Carving Sustainability into Architecture Knowledge Practice

Markus Funke^(✉) and Patricia Lago

Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
m.t.funke@vu.nl

Abstract. In the daily work of a software architect, knowledge is ubiquitous. In addition to technical expertise, architecture knowledge (AK) also requires practical experience in the representation, communication, and management of architectural decisions. However, there is a pressing need to also incorporate sustainability aspects, i.e., capturing decisions towards software systems that are environmentally, economically, and socially balanced in the long term. With this study, we aim to provide a review of AK concepts and their representation and communication from a practical point of view. Having this understanding, we explore where sustainability can be applied in daily practice and how we can address sustainability in architecture processes in the future. The paper presents an empirical study conducted in an industrial context encompassing a questionnaire survey with 32 participants and semi-structured interviews with 15 practitioners; both groups are from a major bank in the Netherlands. Based on the insights gained from combining our findings, we (i) provide a map of applied concepts for communicating and representing AK in a large enterprise, and (ii) discuss potential avenues for carving sustainability into current software architecture practice.



Graphical Abstract.

We thank the architects participating in this survey; Antony Tang, John Klein, and Rich Hilliard for discussions on current state-of-research; and Haben Birhane Gebreweld for executing the questionnaire survey.

1 Introduction

Software Architecture entails the systematic organisation of various software components to construct a comprehensive system [13] and facilitate the reasoning about a given system [3]. In the present day, the life cycle of software has a holistic impact on an enterprise and thus demands consideration on multiple layers of its business operations. To preserve the reasoning and information about the overarching system across all layers, documentation is a crucial part [8]. It facilitates the documentation of knowledge during the design process, i.e., architecture knowledge (AK), enabling the recording of decisions for future reference and leveraging past experience to improve future decisions [1].

Although AK is a well-established field in software engineering, the concept of sustainability has gained significant attention only in recent times [4]. Software sustainability is a multidimensional concept that involves environmental, social, economic, and technical dimensions [17]. Despite the increasing attention given to sustainability, practitioners lack reusable guidelines and consolidated knowledge to integrate sustainability into their daily work [16]. We aim to highlight the critical role of AK across all different layers of architecture and propose the incorporation of software sustainability into existing AK methods. This integration contributes to achieving sustainable development goals.

Following the problem statement we need to bridge the gap between isolated techniques for software sustainability and their application in professional practice. First, we need to know the state-of-practice of software AK and how professionals understand and already support sustainability in their daily work. With these insights, we can either improve current practice on AK or propose recommendations to incorporate sustainability. By combining our findings from an extensive survey in a major bank in the Netherlands encompassing a questionnaire with 35 practitioners and 15 semi-structured interviews, our main **contributions** of this research are twofold: **(i)** we provide an overview of applied AK concepts in a large organisation; **(ii)** we use this understanding to discuss recommendations for applying sustainability.

In the remainder of this section we first provide the background; then we discuss related studies. In Sect. 2 we outline our study goal and describe the applied methodology. Section 3 presents the main findings which are then discussed in Sect. 4. Threats to validity are examined in Sect. 5, while Sect. 6 closes this paper.

1.1 Background

Architecture Knowledge. Throughout this study, we use the definition of AK according to Kruchten et al. [12] as:

Architectural knowledge consists of architecture design as well as the design decisions, assumptions, context, and other factors that together determine why a particular solution is the way it is.

Additionally, in our work we particularly highlight the distinction between AK representation and AK communication. The former, **AK representation**, aims at capturing and preserving knowledge in a certain form. While the latter, **AK communication**, describes how the knowledge is disclosed between involved stakeholders. Figure 1 depicts our view on the interaction of the AK artifacts as a mental model. We can observe that various AK representation methods *express* AK. Such representation *identifies* multiple stakeholders by capturing their concerns and interests in the particular knowledge. While stakeholders have a certain *interest* in the AK, they acquire the knowledge by *using* certain communication methods.

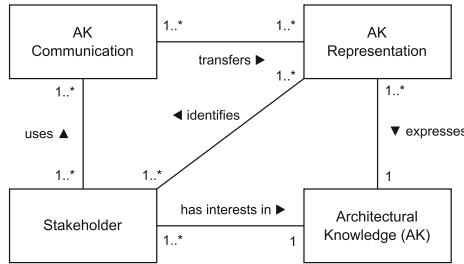


Fig. 1. Architecture knowledge mental model - *Process view*

We can compare our mental-model with the different “AK management philosophies” defined by Ali Babar et al. [1]. The authors differentiate between “explicit and tacit knowledge” and between “application-generic and application-specific knowledge”. While this view can be considered as the *knowledge view*, i.e., from the knowledge perspective, we define our AK mental model as the *process view*, i.e., from the stakeholder perspective. The knowledge view illustrates how the knowledge itself *moves* within the different categories; our process view, instead, illustrates how the knowledge is *utilised* by the actual stakeholders. This view emphasises the utilisation of AK in an industrial context and is considered throughout the rest of our study.

Software and Sustainability. The need for addressing sustainability in architecture has led to various approaches, techniques, and tools for designing [14, 15], evaluating [10], and improving [20] the sustainability of software systems. To the best of our knowledge, however, those emerging approaches appear in isolation without consideration of embedding them in industrial practice.

When we call for balanced sustainability in software architecture, we seek to achieve a harmonious and equitable consideration of the four sustainability dimensions [17] into the design and development of software systems. We recognise that software is a multi-faceted concept which requires a construct of inter-dimensional trade-offs. Those trade-offs demand to be considered at design time, i.e., in software architecture, to align the software with sustainability goals.

1.2 Related Studies

There are few studies that examine the use of AK in professional practice. Malavolta et al. [18] consider architecture modeling languages (AL) as representation tool for AK and conduct a study on the strengths and limitations of these languages by surveying 49 practitioners and identifying their needs and requirements for future AL. Our study goes beyond AK representation and also examines AK elements and communication methods.

In facing AK from the knowledge management perspective (AKM) [1], Capilla et al. [5] determine what industry needs by analysing state-of-the-art AKM tools. The authors raise questions identifying barriers for using those tools and documenting architecture in practice. Even though the focus of this research relies on AKM tools, we are able to build up on this research and reuse for instance certain interview questions for our work.

Dasanayake et al. [7] conduct a case study in an industrial setting to investigate how architecture decisions are made in practice and to improve the decision-making process. The study entails 10 interviews in three companies, revealing that the experts do not follow a systematic approach. The study also finds that the practitioners are willing to adopt lightweight solutions to enhance their decision processes. While the study partially aligns with our goal of comprehensively embracing the entire AK process, it primarily focuses on decision-making.

The studies discussed above have made partial contributions to our research, but none of them have specifically aimed to incorporate sustainability into current practice using AK. Andrikopoulos et al. [2] conduct a systematic mapping study to explore software architecture together with sustainability and find that current research has neglected the holistic viewpoint by focusing on particular sustainability-related dimensions. Lago et al. [16] conduct a more practical study by examining the needs of both researchers and practitioners regarding “architecting for sustainability”. The study uncovers barriers to implement sustainability, such as the lack of understanding among practitioners on how to translate sustainability into their own work.

2 Methodology

To encourage reproducibility and enhance the reliability of our results, we provide an online replication package¹ containing the anonymized data and results related to this paper.

2.1 Study Objective and Questions

The **goal** of this study is twofold. First, we want to provide a review to software architects and the research community about AK and its representation and communication in an industrial context. This understanding helps us in our second goal, i.e., to explore *where* and *how* sustainability can be addressed in the future. We identify two main research questions (RQs) and three sub-questions:

¹ Replication package: <https://github.com/S2-group/ECSA23-AKCR-rep-pkg>.

RQ1 How is AK represented and communicated in practice?

RQ1.1 What are the elements that are represented and communicated?

RQ1.2 How is architecture knowledge represented?

RQ1.3 How is architecture knowledge communicated?

We investigate current professional practice about AK by executing our research together with a major bank in the Netherlands. The industrial context helps us in creating a holistic view on what the state-of-practice is regarding AK. We create a map of a large enterprise and their AK elements (e.g., decisions and principles), representation methods (e.g., diagrams and viewpoints), and communication practices (e.g., corporate platforms and workshops). This helps us in understanding the daily work of architects.

RQ2 How can sustainability aspects be represented and communicated in software architecture?

Building upon the practical insights about AK identified in RQ1 we are able to propose recommendations on how sustainability can be incorporated into daily practice. Based on the additionally uncovered impediments we can establish the current needs in order to achieve balanced sustainability.

2.2 Study Design

To answer our RQs, we organize our research process in four steps (see Fig. 2).

Exploratory Review. In Step (1) we build the necessary understanding about architecture documentation, representation, communication, and AK in general. We talk to three researchers in the field of AK. We enrich those insights by consulting background literature and books (e.g., [1, 3, 6]). This understanding helps us in the subsequent steps to bootstrap our qualitative research.

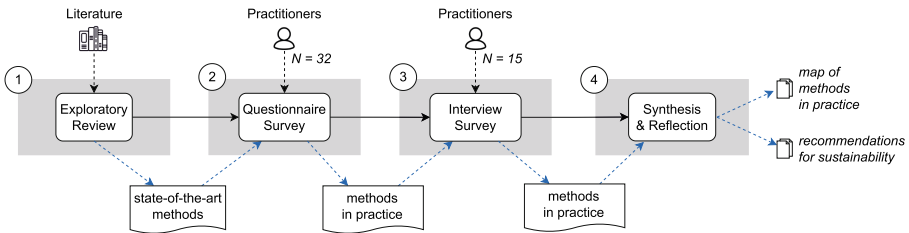


Fig. 2. Study Design

Questionnaire Survey. In Step (2) we construct a series of questions to operationalize our RQs. The questions are related to AK practices, supplemented with commonly used demographic questions. The questionnaire comprises 34 questions in total. With 24 open questions and the rest as guiding and closed questions, we facilitate a candid expression of our participants' unique experiences. The survey guide can be found in the replication package.

Following the design, we first execute a pilot survey with five experts to check the quality of the survey and eliminate potential pitfalls. Then, we determine the main population. Based on the objectives of our research, we determine software architects and similar roles (e.g., domain architects, cloud architects, etc.) as our target population. Based on an internal mailing list and the team leads we generate a draft list of 145 architects. After eliminating redundant names and removing the architects needed for an extensive interview in Step (3) we arrive at a population with 124 architects. The survey is conducted using Qualtrics² and designed to be anonymous to alleviate concerns about judgment or consequences. We reach out to the 124 architects via email and received 45 (39 %) survey responses. After removing the 13 responses that refused consent or dropped after completing the demographics part, the total population counts 32 architects ($N = 32$).

Interview Survey. The aim of Step (3) is to gain in-depth insights from the experts and follow up on the results from the questionnaire. Again, the complete interview guide can be found online. The interview comprises 21 questions in total, all of them designed as open-questions. Most questions are adopted from the questionnaire survey. However, some questions are combined to better fit in an interview setting. It is accepted and especially appreciated if the interview flow lead to other questions or guides the discussion into other directions. The interviewees are selected following purposeful sampling based on two conditions: (i) interviewee has a leading role in the organisation, e.g., manager or head-of, and (ii) interviewee is not part of the questionnaire survey. The leading role allows us to ask more detailed questions of the representatives of an entire group of architects. Further, we do not only select software architects, but rather a broader range of architects including architects with a higher level focus, e.g., enterprise architects or business architects. In total we contacted 21 practitioners, and 15 accepted our invitation ($N = 15$). To determine the length and flow of the interview, we use three experts as pilots. These are included in our final data set as the structure and questions did not change and only a few questions were improved in terms of phrasing.

Synthesis and Reflection. To understand the current practice of AK and uncover potential hooks for sustainability, the final Step (4) synthesises the data gained from both the questionnaire and the interviews.

Data collection and organisation. As the questionnaire is executed online, the collected data is exported automatically into spreadsheets. The answers are not edited. In contrast, all interviews are conducted virtually via Microsoft Teams, audio recorded, and transcribed. The transcriptions are cleaned from emotions and mumbled speech.

Coding and Vertical Analysis. The questionnaire data are pre-coded per question. Those results are used in preparation for the in-depth interviews, e.g., in form of follow-up questions. As both the questionnaire spreadsheets and the interview transcripts follow a similar structure, they are coded together by the first author and validated by the second author following coding techniques for

² QualtricsXM: <https://www.qualtrics.com>.

qualitative data analysis [19]. We start with an initial set of coding categories derived by the RQs (i.e., provisional coding [19]), which are then revised and/or extended (i.e., open coding [19]). We use ATLAS.ti³ as qualitative data analysis tool. The output of this step are the results in Sect. 3. We structure our results according to the identified themes, codes, and sub-codes as shown in Table 2.

Horizontal Analysis and Reflection. To gain further insights from the data, we investigate the responses across our two RQs via horizontal analysis [9]. Our goal is to merge the insights related to AK with those regarding sustainability, and identify potential avenues and gaps for addressing sustainability in AK practice. We reflect on our results by organising all findings in one comprehensive table. The table contains the results from the coding and vertical analysis and the emerged insights from the horizontal analysis. Insights are then turned into recommendations and discussed in several brainstorming sessions between the researchers. To find further practical connections or insights, we additionally discuss this set of recommendations together with two experts from our industrial partner in two informal meetings; one expert related to the higher level of architecture (i.e., Lead of Senior Architects) and one related to the lower level of architecture (i.e., Domain Architect). We summarise our results from this final step in our Discussion in Sect. 4.

3 Main Findings and Results

As in our study design outlined, the coding procedure is identical for both the interviews and the questionnaire survey. After saturation, four themes emerged with 12 codes in total (cf. Table 2). Each code has several sub-codes with more granular results. The complete code-book is available in the replication package. Due to space restrictions, we present and analyse the top sub-codes selected based on their highest frequency - if applicable.

The given frequencies indicate the population who answered a certain question. Not all participants responded to every item in the survey. Items may have been left empty or filled with blanks because the expert was reluctant to answer, did not understand it, or had personal time constraints. If a substantial proportion of respondents did not answer a question, this is mentioned in our analysis. All results and frequencies refer to the data from both interviews and questionnaire; only when significant, we distinguish between the two.

3.1 Demographics

The demographics of our participants is outlined in Table 1. Our entire population comprises 47 participants with 32 from the questionnaire survey and 15 from the interviews. As outlining all different job titles would not reveal strong insights due the variety of titles in a large enterprise, we clustered the participants into either *high-level* or *low-level* architecture. The former, *high-level*,

³ ATLAS.ti: <https://atlasti.com>.

includes jobs like Enterprise-, Business-, or Governance Architect; all operating towards the strategic level of architecture. The latter, *low-level*, includes roles like Solution-, Domain-, or Data Architect; all operating towards the operational level of architecture. For the interviews, we sought balance to complement the questionnaire with its majority on the low-level. We acknowledge that the high-level architecture has a substantial influence on the low-level. Thus, we aim to complement our insights from both levels.

We notice that 42 participants (89%) have engaged in software projects for more than 10 years, with their experience ranging from 11 to 41 years. This suggests that the results obtained were derived from experts who possess extensive and valuable experience gained from a long industrial career. With working for the same company of more than 17 years on average (arithmetic mean), we guarantee that the majority of the experts have profound understanding of the processes in their organisation, leading us to reliable results.

Table 1. Demographics of participants. **I** = Interview; **Q** = Questionnaire; Σ = Summation of Interview and Survey Participants; *years* = Arithmetic mean

(a) Architecture Level				(b) Experience in software				(c) Experience in current organisation			
				Years	I	Q	Σ	Years	I	Q	Σ
				1 - 5	1	0	1	1 - 5	3	6	9
Level	I	Q	Σ	6 - 10	1	3	4	6 - 10	2	8	10
High-Level	7	5	12	11 - 20	3	10	13	11 - 20	6	3	9
Low-Level	8	27	35	21 - 30	7	8	15	21 - 30	3	9	12
				31 - 41	3	11	14	31 - 41	1	6	7
				<i>years</i>	23.5	24.4	24.1	<i>years</i>	15.3	18.0	17.1

3.2 Architecture Knowledge Elements

Most participants provided definitions that reflected our understanding of AK. However, a few participants shared unique perspectives:

“Skills to guide a particular solution landscape in a certain context. This is not a library: it is fluid and keeps developing with every problem I address. Because I have been doing this for a long time, books and trainings are hardly needed, unless big new innovations [...]” (Q_ID-31)

Besides the general understanding of AK, our interest is especially focused on what kind of knowledge elements the experts keep in their professional context and if they could think of any elements which would support their daily work.

Elements. While the majority referred to well known elements as part of their daily work, i.e., (i) architecture design decisions, (ii) standards and guidelines, and (iii) principles, we also found the blueprint (n = 11) mentioned as a driving”element”. Although we have not considered this as a separate element,

Table 2. Results clustered by themes, codes, and sub-codes (*extract*)

(a) RQ1: on AK representation and communication

Theme-1 - RQ1.1 AK Elements	Elements	
	[1.1.1] <i>Design-decisions are a central element</i>	(n = 19)
	[1.1.2] <i>Standards and guidelines are a central element</i>	(n = 15)
	[1.1.3] <i>Architecture principles are a central element</i>	(n = 13)
	Impediments	
	[1.2.1] <i>Missing link between architecture levels</i>	(n = 16)
[1.2.2] <i>Missing business architecture</i>	(n = 5)	
[1.2.3] <i>Missing information about the context</i>	(n = 5)	
Theme-2 - RQ1.2 AK Representation	Methods	
	[2.1.1] <i>Architecture description templates</i>	(n = 45)
	[2.1.2] <i>Architecture diagrams</i>	(n = 34)
	[2.1.3] <i>Guidelines and Standards</i>	(n = 10)
	Standards	
	[2.2.1] <i>ArchiMate</i>	(n = 34)
	[2.2.2] <i>TOGAF</i>	(n = 14)
	Impediments	
	[2.3.1] <i>Knowledge is not captured consistently</i>	(n = 19)
	[2.3.2] <i>Need for more standards</i>	(n = 7)
Theme-3 - RQ1.3 AK Communication	Methods	
	[3.1.1] <i>Face-to-face is used for communicating</i>	(n = 30)
	[3.1.2] <i>Corporate Repositories are a major communication channel</i>	(n = 26)
	[3.1.3] <i>The Architecture Review Board is a central element</i>	(n = 13)
	Stakeholders	
	[3.2.1] <i>Architects are the communication bridge</i>	(n = 35)
	Impediments	
	[3.3.1] <i>Knowledge is like finding the needle-in-a-haystack</i>	(n = 6)
[3.3.2] <i>Tacit knowledge remains implicit</i>	(n = 4)	

(b) RQ2: on carving sustainability into AK practice

Theme-4 - RQ2 Sustainability	Definition	
	[4.1.1] <i>Unbalanced view on sustainability</i>	(n = 21)
	[4.1.2] <i>Comparison with economic costs</i>	(n = 13)
	Daily Work	
	[4.2.1] <i>Sustainability by intrinsic motivation</i>	(n = 19)
	[4.2.2] <i>Sustainability is addressed unconsciously</i>	(n = 11)
	[4.2.3] <i>Usage of dashboards</i>	(n = 5)
	Where and How to inject?	
	[4.3.1] <i>Low level architecture; Bottom-up</i>	(n = 11)
	[4.3.2] <i>High level architecture; Top-down</i>	(n = 10)
	[4.3.3] <i>Architecture Description Templates</i>	(n = 14)
	Impediments	
	[4.4.1] <i>Missing awareness about strategic targets</i>	(n = 28)
[4.4.2] <i>Missing guidance on how to implement sustainability</i>	(n = 9)	
[4.4.3] <i>Lack of knowledge about sustainability</i>	(n = 5)	

as it implies the entire architecture design [3], it is worth mentioning that the design is treated as a holistic AK element by the experts.

Impediments. It appears that there might be a demand ($n = 16$) for a better link beyond the specific architecture documents to (i) the other architecture levels, (ii) the business itself, and (iii) the broader context. We identify that as a need for developing new AK elements that better capture the context and link various architecture views to various stakeholders and the business.

“[...] we tend to develop views that are understood by architects (which gives grip on the architecture), but less meaningful for (other) stakeholders, linking architecture views to (non architectural) views of different stakeholders is now lacking. We tend to speak our own language and not the language of different stakeholders.” (Q_ID-4)

In our interviews we tried to better understand this need. We found that indeed, only on the higher level of architecture documents, there are elements (e.g., diagrams) that explicitly outline the relationship to the business models. At the lower level this connection is not made explicit and the link to the underlying system architecture is lost.

“If we distinguish enterprise architecture and domain architecture what we have in the bank, and then also the lower level [...] system architecture, that linkage is missing. [...] what I would suggest is that we co-create the domain architecture also with the system architecture. So, certain chapters in the template would be created from the domain architect, while others would be created from the system architect. This would lead to one joint-deliverable and enable collaboration” (I_ID-10)

3.3 Architecture Knowledge Representation

Methods. Not surprisingly, 45 experts (95%) referred to architecture description templates and 34 (72%) to diagrams as their method to represent AK. However, most of the experts used their corporate synonyms to refer to a template. For instance, the professionals distinguish between Solution Intents to describe the actual change and architecture of an intended solution on the low-level; and Future State Architecture to represent the envisioned state incorporating strategic goals on the high-level. Nevertheless, all documents are based on one common template. Interestingly, only five experts mentioned views and viewpoints. Recent emerging methods such as Architecture Description Records (ADRs)⁴ or C4-model⁵ haven been mentioned only once.

Standards. Highly related to the methods are standard notations and languages to represent AK. ArchiMate⁶ was mentioned 32 times (72%) as standard architecture modeling language. However, at the same time, especially during the interviews, we also recognised impediments regarding ArchiMate:

⁴ ADRs: <https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions>.

⁵ C4-model: <https://c4model.com>.

⁶ ArchiMate: <https://www.opengroup.org/archimate-forum/archimate-overview>.

“The point with ArchiMate is it’s not easy to understand for people who don’t know the notation [...] I think the knowledge of ArchiMate is really decreasing in the organization [...]” (I.ID-06)

The experts mentioned also methods which cannot be considered as standards ($n = 9$) such as PowerPoint or “Diagram.net” diagrams. This underlines the deep integration of “boxes-and-lines” tools into the daily work.

Impediments. We uncovered that the architecture is currently not consistently represented and captured throughout the bank. This conclusion derives from contradicting results: 19 experts (40%) answered our question about consistency with *no*, while 23 (48%) answered with *yes*. When differentiating between architecture layers, only 16% of high-level architects report a lack of complete consistency across the organisation, while nearly 50% of low-level architects affirm or deny this, respectively. This might be an indicator that the high-level architects are not aware of the lack in consistency on the lower-levels. However, we do also acknowledge the fluid transition between the layers which is in line with the findings from Capilla et al. [5] that consistency in AK is context dependent.

3.4 Architecture Knowledge Communication

Methods. Similar to the results of AK representation methods, the discovered methods in AK communication are in line with our understanding on how AK is communicated. While 26 experts (55%) use their corporate repositories (e.g., Microsoft SharePoint or Confluence) as main communication tool, the majority mentioned face-to-face knowledge exchange in form of scheduled or informal meetings as well as meetings in a workshop setup. The Architecture Review Board (ARB) is used to evaluate all architecture description documents and assess how well they conform to the company’s fundamental principles. We consider the ARB as a central communication element since the knowledge represented and captured in the documents is complemented with tacit knowledge during the review sessions.

Stakeholders. Overall we got diverse answers regarding the stakeholders the architects have to communicate their knowledge. The responses vary from product owner or managers to people who build the architecture, the DevOps team, or the platform team. However, if we consider the role of our expert and match the named roles to their level, in 74% of the cases we observe that it is always *“the level below you, and the level above you”* (I.ID-04). This conclusion confirms the results from Kruchten [11] as they see the architecture role as “communication bridge” between different levels.

Impediments. While we did not encounter any significant obstacles in the communication process, a subset of experts ($n = 6$) expressed the concern AK may be scattered across various repositories. Although only a small number of participants ($n = 4$) reported implicit tacit knowledge, it may still pose a challenge due to the frequent reliance on face-to-face communication as a primary instrument for sharing information.

3.5 Sustainability

Definition. Among respondents, 25 linked environmental factors (e.g., energy efficiency) and 21 linked technical factors (e.g., longevity) to IT sustainability. Thirteen experts connected sustainability directly to economic costs. These findings highlight an imbalanced perspective and limited understanding of the sustainability dimensions. The participants tend to focus either on single dimensions or merge concerns from different dimensions.

“[...] reducing footprints, reducing energy, reducing the use of resources in general - almost equals to reducing costs.” (I_ID-07)

Daily Work. Nineteen experts expressed a personal interest in IT sustainability and intrinsic motivation to address it in their daily work. We reached that conclusion based on participants’ missing awareness about strategic targets pertaining sustainability (n = 28), coupled with frequent mentions (n = 25) of sustainability practices in their daily tasks (e.g., selecting energy-efficient solutions during the design process or incorporating quality attributes regarding sustainability).

“I do something from an intrinsic motivation. So how can I from a data management perspective contribute to the sustainability agenda of [anonymised], maybe the IT sustainability agenda as well?” (I_ID-17)

Interestingly, 15 respondents reported that they did not consider sustainability in their daily work; however, they later mentioned some sustainability-related tasks in their professional routine. The incorporation of sustainability aspects contradicts their awareness of sustainability. This indicates a possibly-unconscious consideration of sustainability.

Where and How to inject? We also leveraged the participants’ experience to understand where and how sustainability could be considered in their daily work. While 11 architects suggest that sustainability should be integrated into the low-level, i.e., the solution design, 10 experts consider the high-level as the right starting level, i.e., enterprise architecture. The most frequent answer regarding the *how* was to embed sustainability into the architecture description templates, i.e., the Solution Intent.

“In the solution intent by using quality attributes. I see sustainability as an aspect of the solution, like security” (Q_ID-12)

Impediments. The majority of experts (59%) are not aware of the sustainability targets in their organisation. This points to a problem in both representing and communicating the two strategic targets on all architecture levels: target (i) *lower the Co2 footprint*, and target (ii) *circular IT assets*.

When asked what would be necessary or what hinders the experts in addressing sustainability, some respondents indicated missing guidance on how to leverage sustainability. This guidance should be either in form of concrete architecture guidelines and standards, tangible strategic goals, and a clear definition of what sustainability means. This reflects the findings from Lago et al. [16].

“We need to have some guidelines, what sustainability requirements we see as a bank or what we have as a bank. [...] So we would need some hooks.” (LID-11)

4 Discussion

In the previous section, results give an overview of AK professional practice in a large-scale enterprise, which helps us achieve our first research goal and answer RQ1. Building on these results, we now turn to our second goal. We conducted a horizontal analysis between RQ1 and RQ2 and identified a list of 14 recommendations. Due to space limitations, we discuss those five that have been prioritised in the two informal meetings (cf. Section 2.2). The complete list is available online. Each recommendation is labeled **R-1** through **R-5** and comes with a boxed example of a specific AK method and the equivalent application of sustainability. The example entails the “as-is” state describing the current situation in the bank⁷; the “to-be” state exemplifies our vision towards achieving sustainable architecture in current practice. All recommendations are grounded on the evidence found during the interviews and questionnaire. We link the recommendations to our findings presented in Sect. 3.

R-1: Repositories contain architecture standards and guidelines developed at a high-level. These standards provide the necessary knowledge about specifications that solutions or documents must conform to, while guidelines represent the recommended course of action. This enables a valuable opportunity to also establish principles and guidelines related to sustainability to guide the architecture design process. We can reuse current practices and ensure that sustainability is not addressed unconsciously, while also providing the required guidance.

R-1: Clear Sustainability Guidelines and Principles

📍 as-is state

Guideline for Databases
Security related:
“D1: Direct links between databases are not allowed in order to achieve isolation.”

Evidence: [1.1.2]; [1.1.3]; [4.2.2]; [4.4.2]

📍 to-be state

Guideline for Sustainability
Cloud related:
“S1: Auto-scaling of cloud application needs to be enabled by default.”

R-2: Given that AK is largely represented in templates and diagrams, we recommend utilizing these templates to add a new chapter dedicated to sustainability assessments and persistently storing sustainability knowledge. Initially, this chapter could include diagrams (e.g., [14]), a sustainability assessment (e.g., [15]), and the adherence to sustainability standards and guidelines (see R-1). By treating sustainability in the same manner as other important chapters such as *security*, it can be effectively integrated into the architecture document. We acknowledge that adding a new chapter may increase the size of

⁷ The “as-is” states are based on real-world examples; however, they have been generalised to avoid disclosing sensitive information.

the template, but if sustainability is a part of a company’s strategy, it requires the same level of attention as other strategic concerns.

R-2: Inject a sustainability chapter into existing templates

<p>📍 as-is state</p> <p>Template outline 1) Stakeholders 2) Solution Architecture 3) Architecture Compliance 4) Architecture Decisions</p>	<p>📍 to-be state</p> <p>Template outline ... 5) Sustainability Assessment 5a) Sustainability by Design 5b) Adherence to Sustainability Standards and Guidelines</p>
--	---

Evidence: [2.1.1]; [2.1.2]; [4.3.1]; [4.3.2]; [4.3.3]

R-3: Given the importance of architecture design-decisions in the current AK practice, we propose capturing sustainability-related design-decisions.

R-3: Design Decisions regarding sustainability need to be captured

<p>📍 as-is state</p> <p><i>“The new process will be migrated without code optimizations due to increased complexity and risks.”</i></p>	<p>📍 to-be state</p> <p><i>“[...] We are accepting the high computation and energy consumption and therefore apply auto-scaling to compensate this trade-off.”</i></p>
--	---

Evidence: [1.1.1]; [4.3.3]

R-4: Based on the current architecture process, it is crucial to address sustainability on all levels and translate individual requirements for each level. However, we face two challenges: (i) the need for new or revised AK elements to better connect the different architecture levels, and (ii) the need for clear guidance on how to address sustainability throughout all levels. To make an immediate impact, we suggest a “bottom-up” approach, starting with the low-level, while also implementing a “top-down” approach on the high-level to ensure lasting effects on future solutions.

R-4: Sustainability needs to be addressed on all architecture levels

<p>📍 as-is state</p>	<p>📍 to-be state</p>
-----------------------------	-----------------------------

Evidence: [1.2.1]; [4.3.1]; [4.3.2]

R-5: Given the critical importance of understanding sustainability in general, we see the repositories currently used (e.g., Microsoft SharePoint) as an opportunity to provide experts with the necessary knowledge. Similar to the e-learning tutorials available on architecture, crucial information about sustainability can

also be communicated. This knowledge transfer is particularly important for members of the ARB, as it is the highest instance assessing new designs.

R-5: Enable architects with knowledge about sustainability

📍 as-is state

E-learning on Architecture:

“This virtual class explains the way of working around Enterprise Architecture and how the organisation works towards a solid architecture.”

Evidence: [3.1.2]; [4.1.1]; [4.1.2]; [4.4.3]

📍 to-be state

E-learning on Sustainability:

“This virtual class explains the way of working around sustainability-aware architecture towards achieving our sustainability targets.”

5 Threats to Validity

External Validity. To increase the adoption of our recommendations, we phrased them in a generic manner for potential reuse, dependent on the AK practices in other enterprises. To further improve the generality of our findings, we clearly describe the context and methodology of our study. Additionally, we selected a diverse population with an average of 24.1 years of experience in software projects and 17.1 years of experience in the company. We also asked participants if they were aware of any AK methods that were exclusively valid in their banking context, and the majority ($n = 31$) responded *no*. Based on this response and their long experience in software in general, we derive that our results are applicable and generally known beyond our specific context.

Internal Validity. To ensure the validity of our survey findings, we acknowledge potential threats due to cultural differences, organizational culture, time and project pressure, and the design of the questionnaire. To address these issues, we carefully designed our survey guides based on existing literature and expert opinions, and conducted a pilot study to test the questions. Additionally, we collected data from multiple sources and triangulated the findings to ensure consistency and objectivity.

6 Conclusion

In this study, we combined the results from an extensive questionnaire and interview survey encompassing 47 architects on various architecture layers. We provided an extensive overview of current AK practice in the context of a major bank in the Netherlands. Based on those insights we propose concrete recommendations on how sustainability can be addressed and integrated. With those recommendations we contribute a major building block towards the overarching need: providing practical guidance on architecting software systems that are sustainable-balanced and creating awareness for a sustainability-aware architecture process in professional practice. In the future, we aim to first deriving insights from other domains, and then apply our recommendations in a real-world setting using an action research approach, with the goal of contributing our findings to both the research community and practitioners.

References

1. Ali Babar, M., Dingsøyr, T., Lago, P., van Vliet, H.: *Software Architecture Knowledge Management*. Springer, Berlin Heidelberg (2009)
2. Andrikopoulos, V., Boza, R.D., Perales, C., Lago, P.: Sustainability in Software Architecture: A Systematic Mapping Study. In: 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE (2022)
3. Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice*, 4th edn. Sei Series in Software Engineering, Addison-Wesley (2021)
4. Calero, C., et al.: 5Ws of green and sustainable software. *Tsinghua Sci. Technol.* **25**(3) (2020)
5. Capilla, R., Jansen, A., Tang, A., Avgeriou, P., Babar, M.A.: 10 years of software architecture knowledge management: Practice and future. *J. Syst. Softw.* **116**, 191–205 (2016)
6. Clements, P., et al.: *Documenting Software Architectures: Views and Beyond*. SEI Series in Software Engineering, Addison-Wesley (2003)
7. Dasanayake, S., Markkula, J., Aaramaa, S., Oivo, M.: Software Architecture Decision-Making Practices and Challenges: An Industrial Case Study. In: 24th Australasian Software Engineering Conference, IEEE (2015)
8. Ding, W., Liang, P., Tang, A., van Vliet, H.: Knowledge-based approaches in software documentation: a systematic literature review. *Inform. Softw. Technol.* **56**(6), 545–567 (2014)
9. Kasunic, M.: *Designing an Effective Survey*. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, Tech. rep. (2005)
10. Koziolk, H.: Sustainability evaluation of software architectures: A systematic review. In: *Proceedings of the Joint ACM SIGSOFT Conference and Symposium on Quality of Software Architectures and Architecting Critical Systems*, ACM (2011)
11. Kruchten, P.: What do software architects really do? *J. Syst. Softw.* **81**(12), 2413–2416 (2008)
12. Kruchten, P., Lago, P., van Vliet, H.: *Building Up and Reasoning About Architectural Knowledge*. In: *Quality of Software Architectures*, vol. 4214, Springer, Berlin Heidelberg (2006)
13. Kruchten, P., Obbink, H., Stafford, J.: The Past, Present, and Future for Software Architecture. *IEEE Software* **23**(2), 22–30 (2006)
14. Lago, P.: Architecture Design Decision Maps for Software Sustainability. In: 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), IEEE (2019)
15. Lago, P., Condori-Fernandez, N.: The Sustainability Assessment Framework. SAF toolkit, Instruments to help sustainability-driven software architecture design decision making (2022)
16. Lago, P., Greefhorst, D., Woods, E.: *Architecting for Sustainability*. In: *EnviroInfo* (2022)
17. Lago, P., Koçak, S.A., Crnkovic, I., Penzenstadler, B.: Framing sustainability as a property of software quality. *Commun. ACM* **58**(10), 70–78 (2015)
18. Malavolta, I., Lago, P., Pelliccione, P., Tang, A.: What industry needs from architectural languages: a survey. *IEEE Trans. Softw. Eng.* **39**(6) (2013)
19. Saldaña, J.: *The Coding Manual for Qualitative Researchers*. SAGE (2016)
20. Vos, S., Lago, P., Verdecchia, R., Heitlager, I.: Architectural Tactics to Optimize Software for Energy Efficiency in the Public Cloud. In: *ICT4S*, IEEE (2022)