European Journal of Information Systems

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjis20

Security by envelopment – a novel approach to data-security-oriented configuration of lightweight-automation systems

Aleksandre Asatiani, Tuuli Hakkarainen, Kimmo Paaso & Esko Penttinen

To cite this article: Aleksandre Asatiani, Tuuli Hakkarainen, Kimmo Paaso & Esko Penttinen (2023): Security by envelopment – a novel approach to data-security-oriented configuration of lightweight-automation systems, European Journal of Information Systems, DOI: 10.1080/0960085X.2023.2217362

To link to this article: https://doi.org/10.1080/0960085X.2023.2217362

Published online: 25 May 2023.

EUROPEAN JOURNAL OF INFORMATION SYSTEMS
https://doi.org/10.1080/0960085X.2023.2217362

Security by envelopment – a novel approach to data-security-oriented configuration of lightweight-automation systems

Aleksandre Asatiani[a], Tuuli Hakkarainen[b], Kimmo Paaso[c] and Esko Penttinen[d]

[a]Swedish Center for Digital Innovation, University of Gothenburg, Gothenburg, Sweden; [b]Work, Organisation and Management Group, University of Liverpool Management School, Liverpool, UK; [c]HR Reporting and Business Intelligence, Wärtsilä Corporation, Helsinki, Finland; [d]Department of Information and Service Management, Aalto University School of Business, Espoo, Finland

ABSTRACT

Organisations' increasing adoption of lightweight automation, such as robotic process automation (RPA), raises concerns about the associated systems' robustness and security, with data-security concerns becoming further accentuated when tools of this sort are deployed for handling of potentially sensitive data. However, literature on designing these tools in a manner mitigating risks related to organisational data security has remained scarce. This paper addresses this gap by presenting a study in which RPA was successfully designed for a process wherein the software robot handles sensitive personal data. Informed by work on the mindlessness of automation, sociotechnical envelopment, and security by design, this empirical study, employing action design research at Wärtsilä Corporation, pointed to three design principles, related to envelopment, access rights, and audit trails. By adhering to these, Wärtsilä created envelopes around the robot that afford the automation's safe operation and processing of the sensitive data. This research advances the theory of sociotechnical envelopment's design and deployment by introducing a novel approach in security by envelopment to elaborate on the security-oriented envelopment of mindless automation agents. The paper also discusses the practical utility of the artefact designed, in terms of both design and evaluation.

ARTICLE HISTORY
Received 16 August 2022
Accepted 19 May 2023

KEYWORDS
Organisational data security; robotic process automation; envelopment; action design research; Lightweight IT

1. Introduction

Robotic process automation (RPA) is a rule-based approach to mimicking human actions in digital knowledge-work processes (Lacity & Willcocks, 2016; Syed et al., 2020). While RPA is a relatively recent arrival, organisations have already gained significant implementation experience and accumulated expertise in how to marshal their army of RPA operations to reap benefits from automating routine operations, advantages such as handling employee data cost-effectively. As RPA permeates ever more business processes, carefully setting boundaries for its implementation is critical. This paper focuses on one boundary of the applicability of RPA, in particular – the handling, processing, and storage of sensitive data. Although many of today's data-security violations could be prevented by having adequate systems in place, most organisations are equipped merely to react to such events *post factum* (Culnan & Williams, 2009; Lin et al., 2022). Numerous pieces of data protection legislation have articulated associated concerns, with the most notable in global terms being the General Data Protection Regulation (GDPR) (European Union, 2016) and the California Consumer Privacy Act (Stallings, 2020). These concerns are further heightened by high-profile incidents wrought amid the recent rise in cyber-attacks (Allianz, 2019) and vulnerabilities in cloud-based environmental configurations (Symantec Corporation, 2019).

Given that RPA is a rule-based automation system, one might be tempted to conclude that its agents are unlikely to deviate from carefully bounded, well-negotiated norms and, hence, be unlikely to introduce data-security risks to the organisation. However, RPA solutions are often not designed for data security by default. Because RPA is implemented atop existing – typically fragmented – IT architecture, the technology is rendered vulnerable to deliberate intrusions and unintended data leaks. Similarly to humans, its software robots are assigned user IDs and granted access to several of the organisation's systems, then extract, process, and store sensitive data from within those systems. By engaging with graphical user interfaces (GUIs), RPA replicates human interactions with the organisation's systems. These robots are unlike humans, though, in that they operate mindlessly and are unable to evaluate and safeguard sensitive data (which are often in rich, unstructured form) while interacting with the various systems and carrying out their tasks. Consequently, RPA deployment often leads to greater risks of data exposure. We argue, therefore, that properly designed, implemented, and configured RPA with the required level of data

CONTACT Aleksandre Asatiani aleksandre.asatiani@ait.gu.se

security represents a topical challenge that warrants greater attention in both academia and corporate practice.

Recent research on deployment of AI models has discussed the approach of sociotechnical envelopment to address the issue of mindless machines (Asatiani et al., 2021; Robbins, 2020). In this approach, the goal is to limit the operational and functional spaces for AI models to minimise the possibility of AI doing harm on the one hand, and, on the other hand, to give greater control to humans. However, existing research has not probed whether a similar approach can be taken for data security. Thus far, research into organisations' practices related to data security has been confined largely to guidelines and frameworks – enforced through the above-mentioned compliance mandates – for ensuring that legal and natural persons alike comply with data-protection and associated terms (Posey et al., 2013). In addition, a large body of work has examined the instruments/metrics used to investigate data-security policies or discuss compliance with fair information practices (Culnan, 2000; Earp et al., 2005). Other studies have investigated the role of "lightweight IT" in addressing security concerns (Bygstad, 2017; Penttinen et al., 2018), with the researchers concluding that, while testing and deployment prove relatively easy, the frequent overlooking of such systems' security aspects might render the organisation more vulnerable to a wide array of risks (Bygstad, 2017; Stople et al., 2017). Therefore, scholars studying RPA have recently called for research into practices employed for managing RPA-related data security (Bygstad, 2017; Moffitt et al., 2018). Still, little guidance exists on how to configure lightweight automation systems (such as RPA) and redesign the underlying work processes such that they address mounting security concerns. These developments prompted us to ask the following question: how can one ensure data security in the act of configuring a software robot to automate work processes that involve processing of sensitive data?

To answer this, we gained access to a unique setting wherein RPA was being developed for work that necessitates processing sensitive data in a manner compliant with new regulations related to the mobility of employees. Thus, we embarked on a seven-month action design research (ADR) project to address how a global knowledge-intensive organisation could implement RPA successfully for a business process that entails processing its workers' sensitive human resources (HR) data to reduce the additional workload stemming from the new obligation created by the EU Posted Workers Directive. We relied on the concepts of envelopment, security by design, and lightweight IT to inform the ADR project. The main outputs alongside the artefact produced were formulation of three core principles for design practice, *security by envelopment* - a novel approach to data-security-oriented configuration, and informative conclusions as to the artefact's real-world utility. Our central contribution lies in articulating the design principles and in understanding how adherence to them facilitates designing RPA in a manner that permits the automation to operate as intended while processing sensitive personal data safely.

2. Background and theoretical foundations

2.1. RPA and the challenges with mindless operation of lightweight automation

Robotic process automation is a technique that follows predefined, well-structured steps in the digitalised process (Lacity & Willcocks, 2016; van der Aalst et al., 2018). Its chief purpose is to automate repetitive and human error-prone tasks at scale, thus improving the efficiency of business processes (Asatiani & Penttinen, 2016; Hallikainen et al., 2018). While the term "robotic" may conjure the image of an intelligent agent comprehensively replacing a human in work activities, RPA operates mindlessly. This constitutes a key challenge for those designing RPA artefacts. Operating in line with a predetermined "playbook" of rules, mindless systems focus on a single purpose, with no awareness of surrounding context (Salovaara et al., 2019). Any apparent intelligence of the system arises from mimicry of true general intelligence.

The mindlessness stems from the fact that the RPA system does not encroach on existing infrastructure (Mendling et al., 2018) and is perceived as a form of lightweight IT (Penttinen et al., 2018; Willcocks et al., 2015). Lightweight systems emerged amid the shift from monolithic to distributed systems, designed to interact with the existing IT architecture without modifying its deep structure (Bygstad, 2017). Again, RPA engages with GUIs rather than back-end application programming interfaces (APIs). Using systems much as human workers do but more quickly, the systems interact with events on a computer screen (albeit a virtual one), mindlessly following the precise steps a human would take to accomplish the task. While lightweight systems are relatively easy to test and deploy, they pose a challenge in that they operate fundamentally outside the established IT architecture: since they operate independently from the core systems, they are less controllable, and these properties entail exposing RPA-controlled operations to various risks, which could critically threaten business operations for which data security is relevant (Bygstad, 2017; Haag, 2015; Stople et al., 2017).

Furthermore, researchers have suggested that realising lightweight automation's full potential necessitates granting RPA access to the business-logic and data-access layers (Hofmann et al., 2020). However,

placing these layers within RPA's reach in its interaction with enterprise information systems' GUIs can introduce potential risks from deeper access. For example, as a mindless system, RPA is incapable of judging whether specific data or system access should be shared with other actors unless the playbook lays down clear rules for this. Information systems' user interfaces and the data available thereby can be "messy", with a host of complexities that render it difficult to create ironclad data-security rules for any mindless agent. Human-controlled operations are often characterised by rough guidance; nothing is fully specified. When dealing with even well-defined routine tasks, humans freely adjust to minor variations in operations without the final outcome changing. However, operations controlled by software robots are characterised by precision and deterministic actions, making them vulnerable to randomness (Salovaara et al., 2019). Hence, RPA can unwittingly expose a process to risks when unexpected changes affect operations, data transfer interfaces between information systems, or systems involving other agents (whether robot or human).

For instance, an RPA that has to move data between two secure information systems requires access credentials for both systems, yet security is not "designed into" this lightweight tool, in marked contrast against software purposefully designed for connecting such systems through an API. When handling sensitive data held in these otherwise secure systems, RPA could expose the process to security risks. In the light of this possibility and, coupled with it, the critical role of data security work within today's organisations – in a world rife with data breaches, system outages and malicious software (Cram et al., 2017; D'arcy & Herath, 2011) – designing RPA processes with security in mind is pivotal for success. Notwithstanding the risks' prevalence and potential severity, we found few, if any, previous attempts to formulate design practices specifically for developing security-focused RPA environments.

2.2. Organisational data security and designing information systems with security in mind

Organisational data security, defined as protection of an organisation's information resources (Anderson et al., 2017), is driven by three sources of pressure: legal requirements, ethics considerations, and pursuit of competitive advantage. Legal requirements mandate organisations' active safeguarding of information. Compliance requirements, typically imposed by local or global legislation (e.g., the GDPR in the EU), exert coercive institutional pressure (DiMaggio & Powell, 1983) on organisations. Ethics considerations often arise when an organisation's actions have negative material effects on consumers or its workers. While organisations might benefit from looser rules on data, other parties may well perceive the results as harmful (Greenaway et al., 2015). Finally, addressing organisational data-security factors can establish a source of competitive advantage. When an organisation signals to the market that its data-related practices apply strict standards that ensure high data security, stakeholders (customers, suppliers, funders, etc.) may see the organisation in a positive light and, hence, be more willing to interact with it and thereby increase the potential for business benefits (Greenaway & Chan, 2005).

For the backbone to our study, we drew from research into security (Lowry et al., 2017) and security by design (e.g., Cavoukian, 2009; Langheinrich, 2001). Our first consideration was that many organisational artefacts directly imply security issues, and organisations do invest in preventing risks (Angst et al., 2017; Chen et al., 2011; Vance et al., 2015). A recent review attests that security is central predominantly in IS studies focused on past incidents and that scholarship should complement the insight produced from these with greater attention to proactive security strategies (Mehrizi et al., 2022). Building on this premise, we propose that more attention needs to be directed to understanding the appropriate design of IS artefacts that require continuous reconfiguration efforts, not merely to reactive design (Mehrizi et al., 2022).

Secondly, given that proactive security control is demonstrably the most viable strategy in the long run (Kwon & Johnson, 2014), we adopted the security by design approach's emphasis on embedding security in organisational initiatives from the start so as to support and safeguard the activities of both individuals and organisations (Cavoukian & Dixon, 2013; Cavoukian, 2009; Langheinrich, 2001). More specifically, we considered seven main principles that characterise security by design: 1) proactiveness; 2) instituting "secure by default" policies, such as need-to-know principles; 3) embeddedness in the design; 4) a positive-sum approach (i.e., accommodating all stakeholders and resolving conflicts in pursuit of a win–win outcome); 5) end to-end security; 6) visibility and transparency; and 7) respect for the users' data and protection of said data (see Cavoukian & Dixon, 2013). While high-profile organisations such as IBM, Deloitte, and Ernst & Young are embracing security by design, research addressing how best to apply these principles remains scarce. To advance our understanding of this proactive approach to security in the context of designing an RPA artefact, we underpinned our work in the principles of security by design, to support security by default.

2.3. Envelopment as a possible solution to the mindlessness problem in the data-security context

Our search for a way of proactively designing secure RPA systems led us to the concept of envelopment.

This refers to an approach recommended for safeguarding the processes managed by mindless agents within a complex network of distributed information systems (Asatiani et al., 2021; Floridi, 2011; Robbins, 2020). We argue that envelopment could be the primary thrust of attempts to design RPA in a secure manner. Often, the concept is most readily understood by reference to its origins in early work in the physical-robotics field, where it refers to "the set of points representing the maximum extent or reach of the robot hand or working tool in all directions" (Scheel, 1993). Here, maps of the factory floor employ stripes or shading for clear delineation of the environment within which the robot is allowed to operate, the work envelope. Envelopment entails ensuring that the number of states of any control mechanism (here, the robot) is larger than the number of environmental states in which it operates. That is, if made to operate in an environment whose complexity exceeds the robot-internal capacity for comprehending the surroundings, a robot poses a risk to its surroundings. With envelopes – areas that no other actors will enter – one can guarantee that the robot's physical environment is simplified sufficiently (i.e., its number of states reduced enough).

Extending this concept to the digital realm, where most parameters for the envelope are not physically specified, one can envelope a digital agent by controlling that agent's operation playbook (i.e., its set of instructions) by setting boundaries to the inputs from which the agent acts and establishing solid awareness of the agent's functions and outputs (Robbins, 2020). In practice, this involves carefully curating the access the RPA agent has to specific information systems, bounding the inputs to the agent and ensuring that all outputs of the mindless operation serve the intended objectives. A carefully crafted envelope puts all the undesirable outcomes out of reach, rendering it impossible to, for example, convey private data to unauthorised third parties. The envelope can also contain sensitive data that is necessary for the RPA to produce its output but this sensitive raw data should not be accessible to the recipients of the output and should be discarded within the envelope instead (Figure 1 provides a simplified depiction of the envelopment concept and its potential relation to RPA).

Simultaneously, the envelope should not impede critical operations of the agent or negate the efficiency gains from automating the tasks. This simultaneous, co-existing aim to achieve both instrumental (i.e., efficiency requirements) and humanistic (i.e., security and privacy) outcomes (Sarker et al., 2019) points towards the need to look at this phenomenon through the lens of sociotechnical approach. Indeed, recent research on envelopment of AI has emphasised the sociotechnical nature of the approach (Asatiani et al., 2021), suggesting that one needs to rethink technological design in tandem with social aspects of organisational processes (e.g., how humans perform certain tasks). Given that RPA is designed to precisely mimic humans' actions, adopting a sociotechnical approach to its configuration is even more important. By configuring RPA to perform a manual task, it is essential to also rethink how the task itself should be adjusted given the advantages and drawbacks of the mindless agent.

In the light of this background, we find envelopment suitable for application in RPA. We are not the first to argue that envelopment can function as a mechanism bringing mindless agents under control when security and accountability are crucial (Asatiani et al., 2021), with some literature, based on empirical investigation (Asatiani et al., 2020) and conceptual

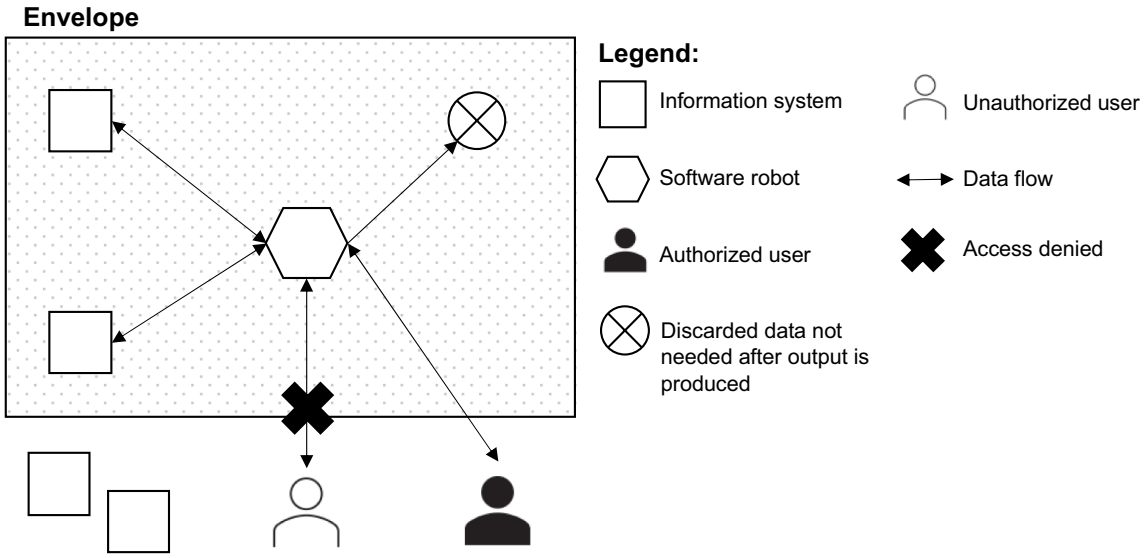

Figure 1. Envelopment concept.

efforts (Robbins, 2020), also offering practical recommendations and examples of the envelopment mechanism. However, research has left it unclear how exactly envelopment should be designed for knowledge-based tasks. Moreover, there is limited understanding of whether envelopment suits security-oriented applications.

3. Method

The findings presented here emerged through our seven-month research project aimed at creating an automated process that uses RPA to gather the data required for submitting EU posted-worker notifications[1] and thereby reduce the additional workload stemming from the new Posted Workers Directive. In the project, carried out in June to December 2018, we employed building – intervention–evaluation cycles as explicated in writings on the ADR approach (Sein et al., 2011). An ADR design allows one to carry out immersive industry-based projects (Mullarkey & Hevner, 2019) wherein researchers and practitioners together shape the artefact over the course of the project (Haj-Bolouri et al., 2018). Because the emphasis in ADR is on the act of building the artefacts through the BIE cycles (Sein & Rossi, 2019; Sein et al., 2011), this approach is well suited to pursuing the objectives behind our research. We chose an ADR framework for building the relevant software robot because we wanted to work in close cooperation with the company developing and deploying the robot, for activities whereby the research and practitioner community contribute to each other's understanding. We had access to a unique setting wherein RPA was being developed for work that necessitated processing of sensitive data. This represented significant potential in that, to the best of our knowledge, the process of configuring automation for such a process had not been studied before.

3.1. The research setting

The case organisation is Wärtsilä, a Finnish multinational corporation and a global market leader in innovative technologies and life-cycle solutions for marine and energy markets. Wärtsilä has 17,500 employees, across 80 countries, with operations on 200 sites. This industrial corporation's operations naturally include selling physical products but also lifecycle solutions inclusive of maintenance and upkeep. Therefore, Wärtsilä employs numerous travelling engineers to provide these services, which encompass providing on-site support, overseeing multiple projects, managing regional development and advising on construction projects, all of which require visits to various sites. Since the engineers perform work that falls under the EU's Posted Workers Directive, notifications must be filed in relation to each visit. For Wärtsilä, the associated significant administrative burden brought challenges and also an opportunity to start automating the gathering of data in response to the need for time-consuming, high-volume notification work.[2] Such endeavours dovetail with the company's strategy: to date, the teams within Wärtsilä's various business divisions and functions have automated more than 400 processes, which run the gamut from simple to quite complex operations (UiPath, n.d.).

3.2. Data collection

To set up the ADR study, we formed three teams, composed of researchers, practitioners, and end users. The research team had four members: three full-time researchers and a person responsible for the practical implementation of the automated process (Purao et al., 2013; Sein et al., 2011). The responsible person acted as the main channel of communication with the company, was part of the company's International Mobility team (within the Global HR division) and was responsible for articulating the requirements for creating the notifications and the process (its design, implementation, and automation). The practitioner team consisted of this person and two colleagues at Wärtsilä. The research and practitioner team together constituted what we refer to below as the ADR team (Sein et al., 2011). Finally, the end-user team was formed of system experts, personnel at the company's RPA centre of excellence and potential internal end users of the new RPA.

The data collection included regular formal meetings with various Wärtsilä employees, who gave ongoing input based on their areas of expertise. The meetings were held face to face (whenever possible) or via a video connection. In total, there were 22 meetings over the course of the seven-month ADR project to discuss particular aspects of the project with the various stakeholders. Extensive notes were taken on all of them. In December 2021, 3 years after acceptance of the artefact's release version, we conducted two interviews with key project stakeholders to probe retrospective reflections on the project and on the artefact's production use. Finally, 1 year after those, we held three "applicability check" sessions (Iivari et al., 2021), for critically evaluating the reusability of the design principles derived from the project.[3] Table 1, below, outlines the data-collection process, mapping the BIE cycles (Sein et al., 2011), events, meetings, and data collected to the temporal flow of the empirical study.

The ADR project began with gathering data on the overall design of the reporting process. To this end, 10 employees were identified who would be responsible for submitting the EU posted-worker notifications and ascertaining which solution is best for this (e.g.,

Table 1. The main events during the project and the data sources.

Month and intervention cycle (if applicable)	Event	Data sources
June 2018	Country-specific mapping of the requirements set by the Posted Workers Directive	Meetings with service co-ordinators.
July 2018	Deciding on a centralised rather than a decentralised governance model	Meetings with candidate groups of employees for handling notifications Briefing of local HR contacts
August 2018 (first BIE cycle)	The decision to divide the software robot into two parts, with the component accessing human resources SAP system information to run on a separate server	Meetings with the human-resources SAP system's owner and other productowners
September 2018 (second BIE cycle)	The decision to use a database to store data (the software robots would utilise this database also to pass certain information between systems)	Meeting with RPA experts
October 2018 (third BIE cycle)	The decision to use Power BI for sharing information on travelling engineers with local HR contacts	Meeting with Power BI and Qlik Sense experts
December 2018	Acceptance of the artefact	Presentation of the RPA pilot artefact
December 2021	Retrospective reflection on the project and on the artefact developed	Interviews with two key stakeholders of the project (internal evaluation)
December 2022	Critical evaluation of the design principles' reusability	Applicability checks with RPA practitioners (external evaluation)

producing the smallest administrative burden): two general managers and eight team leaders, all of whom send service engineers to several EU countries. Follow-up on these meetings entailed consulting three managers (from a shared service centre, a servicework site, and a regional HR entity) for input to ascertaining who could allocate the resources needed for the notifications. One question at this point was how centralised the governance model ought to be. The shared service centre preferred a relatively consolidated approach (with a single team handling the notifications for the whole company), while the service-work site advocated a decentralised method (in which the host country's HR representative completes the notifications for all employees travelling to that country). Decentralisation offered the benefits of exploiting the local representatives' familiarity with local laws and processes, while centralised governance promised greater economies of scale.[4]

After this stage, meetings were held with the system experts, the relevant HR system's "owner", and a concept-owner for data warehouses (who represented various applications from which the data for the notifications could be retrieved). The main data needed for the posted-worker notifications were details of the engineers themselves and of the work they would be doing abroad. At this point, it was unclear which sources of data within the organisation would be appropriate. For identifying suitable sources, meetings canvassed a wide range of internal experts, each in charge of diverse information systems. Our aim was to find a combination of systems that would include all the required data items while simultaneously being compatible with RPA. Five source systems were reviewed, and it was decided that the project would use three of these: the CRM system, supplying data on the duration and type of work being done; an SAP system used by HR staff for handling employee information; and another SAP system, containing details of the subsidiaries sending workers abroad. In our meetings with the experts, we presented the necessary data items, asking the experts to indicate which of them could be collected from the respective systems. We also considered several other factors, such as the time required to integrate RPA into the system, costs of integration, and the reliability of the data.

Once the source systems had been selected (the CRM system, the HR-specific SAP system, and the other SAP system) and initial development of the software robot had started, the project personnel arranged meetings with three RPA experts (a solution architect, an expert in RPA solutions, and a development manager), to elicit their feedback on our initial designs and solutions. Because Wärtsilä employees showed keen interest in RPA at the time, there were many enthusiastic RPA experts willing to provide support and feedback for the project. Proceeding from the feedback on the RPA process, the practitioner team compared several products, with a view to creating a dashboard for notification-related data. The idea was to allow the local HR personnel to see foreign employees stationed locally and also the local workers posted abroad. Dashboards were created for two systems (Power BI and Qlik Sense), with the development informed by input gathered in two meetings with an expert in both systems (a worker familiar with their usability and cost structures). We knew that creating these dashboards and the associated reports would be possible through both solutions, but we wanted to delve into the costs involved and see which solution would be more suitable for sharing these bespoke reports with roughly a dozen local HR contacts.

Finally, the two interviews held with key Wärtsilä stakeholders in December 2021 probed retrospective reflections on the project, sought input on the subsequent deployment and requested evaluations of the ultimate efficiency of the artefact developed (Appendix A summarises the resulting Wärtsilä-internal reflections), and the December 2022

applicability checks conducted with RPA experts assessed the design principles' generalisability (the outcomes from these external evaluations are presented in Appendix B).

3.3. Formulation of the problem

The impetus for the ADR process came from Wärtsilä's practical problem of having to file EU posted-worker notifications to fulfil legal obligations. The company's initial response was to hire two people to design a process for handling these notifications. The EU Posted Workers Directive has been subject to increased attention in recent years – before the 2014 enforcement mandate, companies did not have to submit notice to the host country when sending employees to work there, but this became a legal requirement once the directive was transposed into most countries' national law, in 2017. Since then, EU-based companies have been developing processes to handle the notifications smoothly. Thanks to the peculiarities of each country's implementation of the directive, the process of filing notifications is a complex one for companies that regularly post employees to multiple EU member states.

The research team joined the project once Wärtsilä had decided that the process would be automated. At this point, the company had already specified the practical situation, so the research team proceeded to devise a general problem formulation to serve as a basis for the emergent artefact (Mullarkey & Hevner, 2019; Sein & Rossi, 2019). As the research team entered the field, it was considered important to communicate the value of the general knowledge to be yielded by the research, to improve the project's practical outcomes, as suggested by Haj-Bolouri et al. (2018). In June 2018, the practitioner team started to research each host country's local implementation of the Posted Workers Directive. At this point, the research team discovered that EU countries were at quite different stages of implementation with regard to the notification process: Finland had an online form that could be filled in rather easily, without any need for specific data on individual employees, and submitting the form did not require creating an account on the Website hosting it. France, in contrast, was using an online portal that necessitated the creation of an account, and the notification itself had to contain substantial quantities of personal employee data. Greece, in turn, required each notification to be sent by post, and some countries, such as the Netherlands, had yet to implement the directive at all.

After researching the country-specific implementations, the practitioner team started mapping systems that could gather the required data. For certain data, only one or two specific systems stored the relevant details. For instance, the CRM system and the SAP system were the only sources for data on employee salaries and the customer for whom the work would be done. The practitioner team reviewed various sources in efforts to find the minimal ("lightest") combination of systems for gathering all the required data. In the end, the team decided to obtain most of the data from the CRM and SAP systems, while storing other, static data in Microsoft Excel sheets. Among the latter items were ones that change only rarely, such as the company address. These were easier to retrieve from an Excel file than from a third system.

Next, the practitioner team started analysing how many international assignments were being made, to get an idea of the volume of this essential travel. To do so, the team consulted the CRM system, which covered the assignments of the service engineers, who were the centre of the team's focus since they were the largest group of travelling employees necessitating posted-worker notifications. From 2 years' worth of intra-EU travel reports, the team ascertained that international postings for this set of employees involved more than 1,000 trips per year. Depending on the destination country, anywhere from 10 to 30 min was spent on gathering the data needed for a notification. Since the estimated number of trips did not cover everyone whose work was subject to the notification requirement, it became clear that Wärtsilä would need at least one full-time worker to take care of notifications for the full volume of travel involved.

Proceeding from this information, the joint ADR team began considering ways to alleviate the administrative burden. As handling of EU posted-worker notifications was classed as a non-core process, Wärtsilä hoped to automate as much of the process as possible in the ideal case. Still, there were limits to the resources available for such an automation project, given the process's peripheral nature (principally, developing and integrating bespoke back-end automation was out of the question). Wärtsilä had already started developing in-house resources for implementing RPA in multiple processes across the organisation. Accordingly, the practitioner team could get in-house support for the implementation from the developers, and, since RPA deployment was a top-management-driven initiative, obtaining permission for its use in the project was fairly easy. However, the RPA in place, while efficient, did not meet all of the security requirements out of the box. For automated handling of employees' sensitive data, the process had to be accepted by multiple parties and purposefully designed in such a way as to ensure data security and accountability. This included ruling out unauthorised requests to the software robot, guaranteeing that the robot's actions could all be traced and ensuring the ability to identify and rule out any possible data-security risks brought by the process.

For this ADR study, the problem identified belongs to a general class of problem referred to as system configuration. Configuration problems are distinct from other classes of problems (such as diagnosis, repair, and control) in being centred on the construction of well-structured systems (Clancey, 1985). At Wärtsilä, the final artefact needed to be configured within a context of specific constraints pertaining to the security requirements connected with the artefact. Because the artefact to be configured was a software-robot system, its development had to consider characteristics peculiar to systems of this kind, traits that have the potential to create unique problems. They are unable to judge in a mindful manner, which data elements are sensitive and how to stay within the regulatory boundaries set by the GDPR when handling those elements. Hence, while steps taken by humans can be translated directly into actions by an RPA artefact such that the artefact can perform those steps in the process successfully, the resulting artefact does not automatically meet security requirements. The specific practical problem identified was that of configuring a software-robot system not only to emulate human interaction with information systems and data but also to emulate human mindfulness, so as to meet the security requirements for the system. The research team framed the evaluation and tackling of this problem via envelopment and security by design – proactively embedding security in the system design. The process of designing and evaluating the artefact was guided by three meta-requirements that emerged from examining Wärtsilä's practical challenges in the light of the research streams informing our project:

R1: The automation artefact should proactively protect sensitive data as it accesses and processes said data.

R2: The automation artefact should be resilient to unexpected environmental changes.

R3: The artefact design should maintain acceptable levels of functionality and efficiency.

4. The intervention cycles

After the ADR team jointly formulated the problem and the meta-requirements, the practitioner team created the initial version of the software robot (v. 1) and proceeded with the BIE process cycle (Sein et al., 2011). This involved three iterations of the cycle, resulting in versions 2, 3 and 4 of the artefact (see Figure 2). The target was to create an artefact that would gather the data required for the EU posted-worker notifications and send these to the employees designated as responsible for submitting the information to the competent authorities.

4.1. The human process and version 1 of the robot

Before development of the first version of the robot could commence, the reporting process had to be mapped from a human employee's perspective (see Figure 3). In the process diagrammed, the employee reviews a list of upcoming trips and identifies which of these require a notification. Then, that employee submits the relevant CRM ID to the robot, which gathers the required data

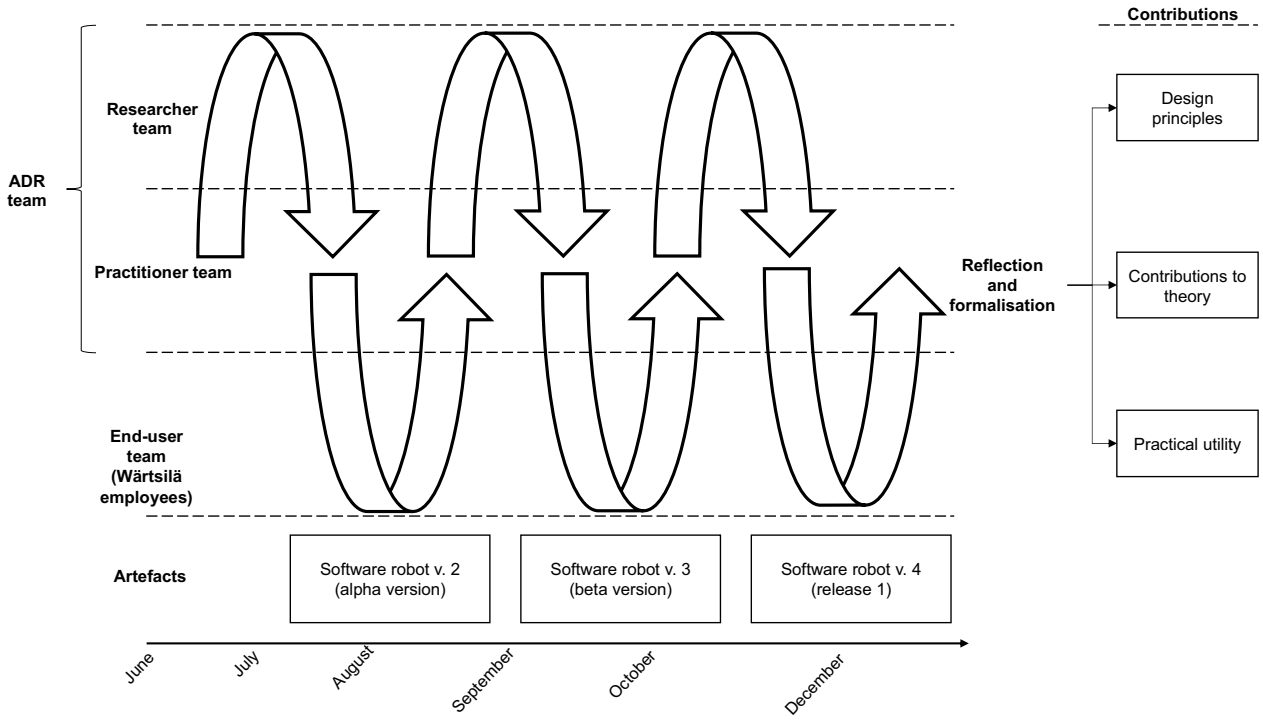

Figure 2. The building–intervention–evaluation process.

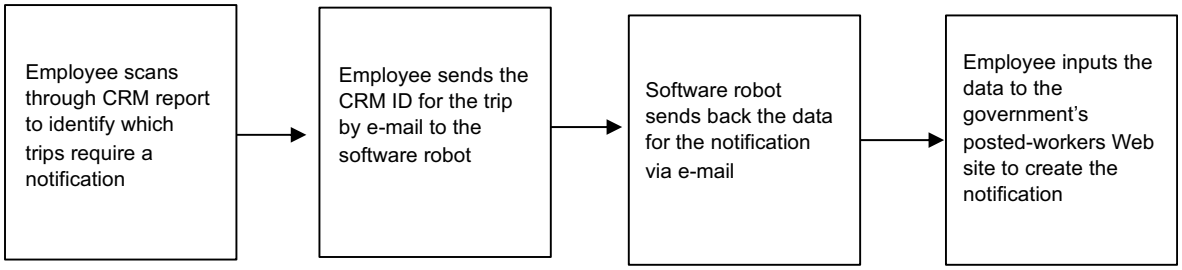

Figure 3. The reporting process from the employee's perspective.

and sends said data back via email. On this basis, the employee then enters the data on the host country's posted-workers online site to create the notification or sends the information by post, as required. After this, the employee monitors the trips for which notifications have been filed, in case of changes such as adjustment to an assignment's end date.

Prior to the development of the first version of the software robot, the source systems were mapped and the data items needed for the notifications were identified. The next step consisted of deciding how the software robot should be triggered, the order in which the robot should access the source systems, and how it should return the data collected to the person submitting the notification. The RPA software used was called UiPath.[5] Since the software robot ultimately runs in a virtual environment and human action identifies the trip for which the software robot proceeds to gather data, the developers chose email as the mechanism for invoking the UiPath robot. A CRM ID from Salesforce was chosen as the means of identifying the trip on which the robot would collect data, so the human initiator's email message would have to include this datum. From the data in the CRM system, the software robot should be able to identify the company for which it is to gather the company data from the corresponding SAP server. From the CRM system, the robot should obtain the employee's SAP system ID, which would then be used to gather the data for that employee from a second SAP system, used in HR operations. Accordingly, the first draft of the design for the software robot was fairly straightforward (see Figure 4).

In this design, the robot is triggered by the above-mentioned email containing a CRM ID. The software robot reads the CRM ID from the email and opens a Web browser to go through the CRM system and gather all the data items needed from it (34 in all). After this, the robot logs into the SAP system, from which it retrieves the address data and the sending company's VAT code and business ID (10 data items), using the employee's company code as retrieved from the CRM system to find the correct company in the SAP system. The next step is login to the HR division's SAP system, from which the robot gathers the employee's personal data (17 data items). The information includes such items as the date of birth, home address, and salary that match the employee's personal SAP system ID obtained from the CRM system. Once the robot has gathered all the data needed, it checks the CRM data for the country the employee will visit and then creates an Excel file containing the data items necessary for that specific country. Finally, the Excel file created is sent back to the person who invoked the software robot.

4.2. The first BIE cycle

Work in the first BIE cycle concentrated on presenting the initial version of the software robot to the end user team, gathering that team's feedback and identifying the gaps between the requirements and the current state of the system. By design, the work behind the first version of the robot focused on automating the task in the most straightforward manner possible (to meet R3) while largely de-emphasising security-related meta-requirements. The practitioner team's objective was to expose end users to a functional

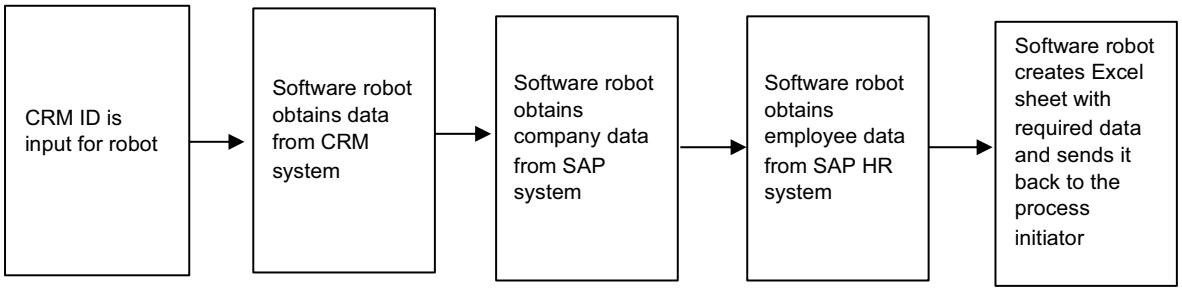

Figure 4. Software robot v. 1.

prototype so as to obtain feedback for further development and identify potential security issues. At the end of the first cycle, the research team evaluated the results alongside the practitioner team, who proceeded to build the second version of the software robot.

4.2.1. The feedback on software robot v. 1

The artefact was presented to the product-owner for the human-resources SAP system before anyone else, since the software robot could not receive access to the HR SAP system without her approval. The actions designed for the software robot within the SAP system were streamlined slightly by means of reports for gathering the necessary data more readily than was possible with the items' piecemeal collection via several transactions in the SAP system. A more significant output of this meeting was the policy that, were the robot to use the HR SAP system, it would have to run in a separate, closed environment. This was because giving the robot full access rights to that system exposes data intended to be visible to authorised HR employees only. Typically, multiple software robots and processes can run in a single environment, with all admin-level users having access to the virtual environments so that they can maintain the robots and debug them if errors occur. Since this software robot would be accessing sensitive data, data security could be compromised if many people were able to just enter the virtual environment and use the robot's access rights to view data that they as humans are not allowed to access.

The meeting uncovered another issue too, closely connected to the need for a separate environment: anybody who knows the email address of the software robot could trigger it by sending a CRM ID and thereby receive the information then gathered for that ID. There would need to be controls on who can access the software robot and trigger it – that process had to be redesigned so as to not only handle the operations with the human-resources SAP system in a separate environment but also incorporate some sort of access control ensuring that only authorised persons can trigger the robot.

4.2.2. Evaluating the results and building v. 2 of the software robot

The end-user team's feedback on v. 1 indicated that the robot performed the basic task of gathering and compiling the data adequately, bar minor efficiency issues. As was expected, the artefact did not satisfy security-related requirements; this was manifested in relation to two major issues: 1) the robot's full access to all systems containing sensitive data, which should be restricted to authorised personnel, and 2) lack of means to control who may access the sensitive data processed by the robot. Security features to address such issues had not been part of the human process, since the human operators were less likely to share sensitive data with outside parties inadvertently and do not store any of those data in separate documents external to the CRM and SAP systems. While equipping the artefact with an ability to access all the required systems without imposing the burden of security elements did afford simplicity and efficiency, which are desirable in an RPA artefact, such a design philosophy would be at odds with the fundamentals of security by design and would violate practical security requirements.

For the v. 2 design, the ADR team focused on fulfilling the meta-requirements related to security (R1 and R2), while de-emphasising R3 for the time being. In practical terms, this entailed addressing the two security issues highlighted above. The first change was to make the email reading component more secure by design through removing ambiguity as to who can initiate the process. Figure 5 shows the process chart for the email checking loop, which

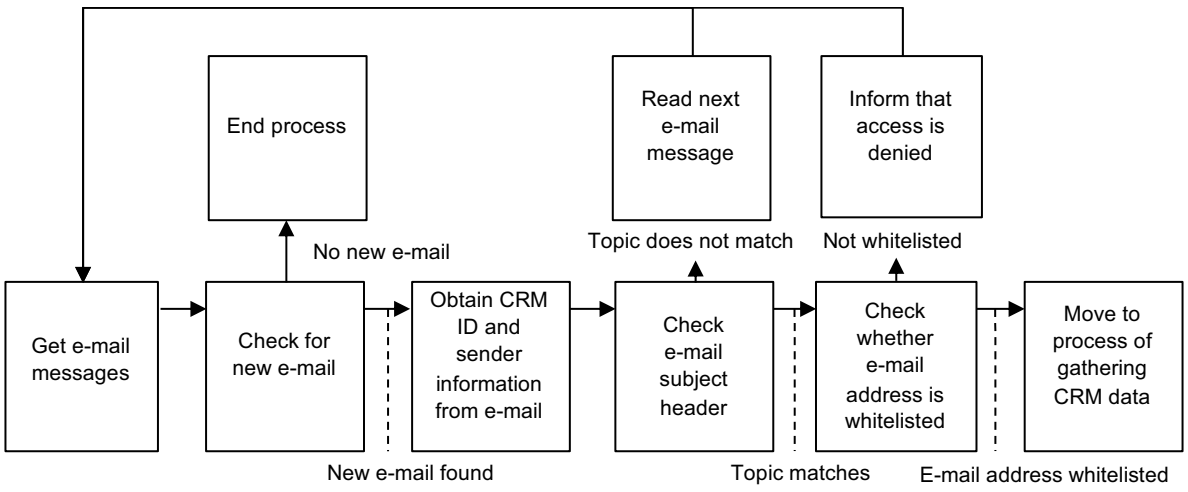

Figure 5. The e-mail reading process.

sits within the CRM ID input box in Figure 4 (above), the box representing the robot's trigger mechanism.

In this process, the software robot first checks for new email, and the process exits if no new messages are found. If there is a new email, it gathers information from the message, such as the CRM ID indicated, the Subject header and the sender of the email. Once it has done this, it checks the subject line to verify that the email was indeed meant to trigger the robot; that is, the robot looks for "Posted Worker" and bypasses all email not featuring that term in the subject field. This minor check adds another layer of control to the robot and, more importantly, guides the robot to read only the relevant messages – if the Subject header does not contain this string, the robot returns to the beginning of the process and reads the next message. When the subject of the email does match, the robot then checks the sender's email address against a list of whitelisted addresses coded into the software robot. If the address is not on the whitelist, the robot sends the originator an email message stating that access to the software robot from that address is not permitted.

The second change was guided by the idea of envelopment, under which any robot accessing the HR SAP system is confined to a separate environment. There was a practical problem, however. This part of the procedure could not simply be made independent of the rest of the process, since certain data must be transferred between the two environments. For tackling this issue, the team decided that the software robot gathering CRM system data could create the Excel file and populate the necessary fields, then send both the employee's SAP system ID and the semifinished Excel file to a second robot; from here, the second one could fill in the rest of the employee information and send the file on to the original request-submitter.

The resulting RPA (v. 2, depicted in Figure 6) featured both the new email reading process, adding a layer of security, and the division of the process between two environments, with two separate software robots. To add further security to the process in which the first robot gathers the data needed from the source systems and conveys the data to the second robot via email with the attached Excel file, the second robot employs the same email reading process as the first and takes the CRM ID from the email. With this ID, it gathers a few data items from the CRM system, such as the employee's SAP system ID. The latter enables it to move on to the human-resources SAP system and commence gathering the employee's personal information from there. Another change in design was to make sure the employee's salary data are not gathered unless required by the relevant country. This makes the process faster in general, thus helping to meet the efficiency requirement, and avoids unnecessary retrieval of the most sensitive data. As described above, the final steps involve the second robot filling in the rest of the rows in the Excel sheet and sending the resulting file back to the person who triggered the robot.

4.2.3. Reflection on the first BIE cycle

In the first BIE cycle, the ADR team learned that some aspects of the human process's translation to RPA and the corresponding simple artefact design introduced security issues. The prototype (v. 1) met the functionality and efficiency requirements (i.e., R3) but demonstrated major data-security risks in its operation. The iteration for v. 2 refocused the design, towards addressing the security meta-requirements and thereby

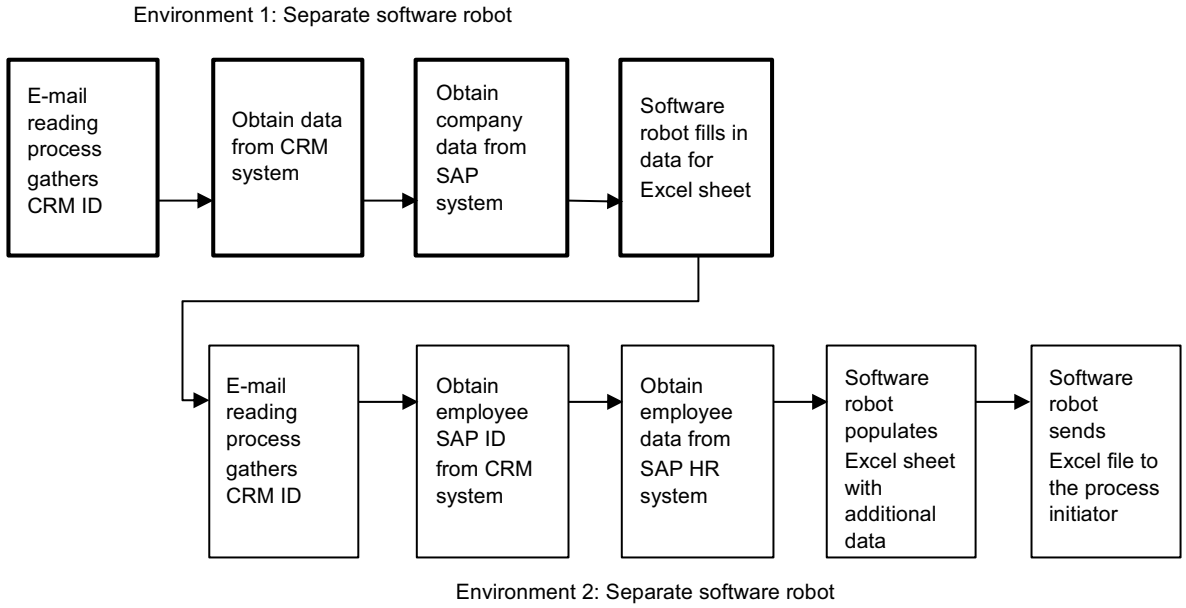

Figure 6. Software robot v. 2 (alpha version).

ensuring proactive protection of sensitive data, alongside the system's resilience to unexpected events (i.e., meeting R1 and R2). The core principles of security by design and envelopment guided the solutions to the security problems that reared their head. Here, the artefact design incorporated clear access control, preventing non-authorised actors from accessing the data, and created separate envelopes for robots interacting with the CRM and HR systems. Both of these actions served to reduce randomness in the mindless robot's operation environment and increase the default security of the artefact. While such design choices run counter to the rules for efficient design, which constituted the paramount consideration behind v. 1, some of the efficiency lost could be regained by limiting the data gathered for each request, a move that further enhanced security too.

4.3. The second BIE cycle

The second version of the robot was presented to RPA experts at Wärtsilä for feedback, with the trigger for the second cycle being to solicit review from a technical perspective and gather improvement suggestions, for rendering the whole process faster and more efficient. Addressing the issues identified during the first cycle had introduced greater complexity to the artefact, and the robot's endto-end operation now took around 3 min. At the same time, further complexity could usher in additional security issues and undermine system transparency. Where the first BIE cycle had focused on identifying and addressing the most glaring issues, the second one offered an opportunity to refine the artefact.

4.3.1. The feedback on software robot v. 2

Version 2 was presented to both the human-resources SAP system's owner and the RPA experts. The most significant improvement suggestion that the end-user team offered in the second cycle was to stop storing data locally in Excel files and start using a database instead. The former process was deemed overly complex: software had to create an Excel file; populate the appropriate cells; attach the resulting file to an email message; and send that message to the next robot, for opening, completion, and passing on to the triggerer. Hence, employing a database could be expected to yield numerous security and speed benefits: The argument with regard to data security was that using a database rather than sending Excel files means that the data are not stored locally. In addition, there is far greater control over the accessibility of data in a database, since only a few key users need to be given access to where the information is stored. Data could be kept in a database temporarily, then get wiped from it when the software robot is finished compiling the material.

4.3.2. Evaluating the results and building v. 3 of the robot

With this iteration, the feedback at the start of the cycle was more focused, prioritising incremental improvements to polish the design from the v. 2 artefact. There were two main issues identified on the basis of said feedback: 1) simplifying the design of the robot, to mitigate the complexity introduced over the course of the first cycle while retaining the system's security gains, and 2) further improving security, with a focus on the way of handling the data. The main fresh items considered for refinement of the robot and process design were the new way of reading email, guaranteeing greater control over who can trigger the robot, and the use of a database to transfer data between two software robots. The overall layout of the software design did not change at all in this iteration of the cycle.

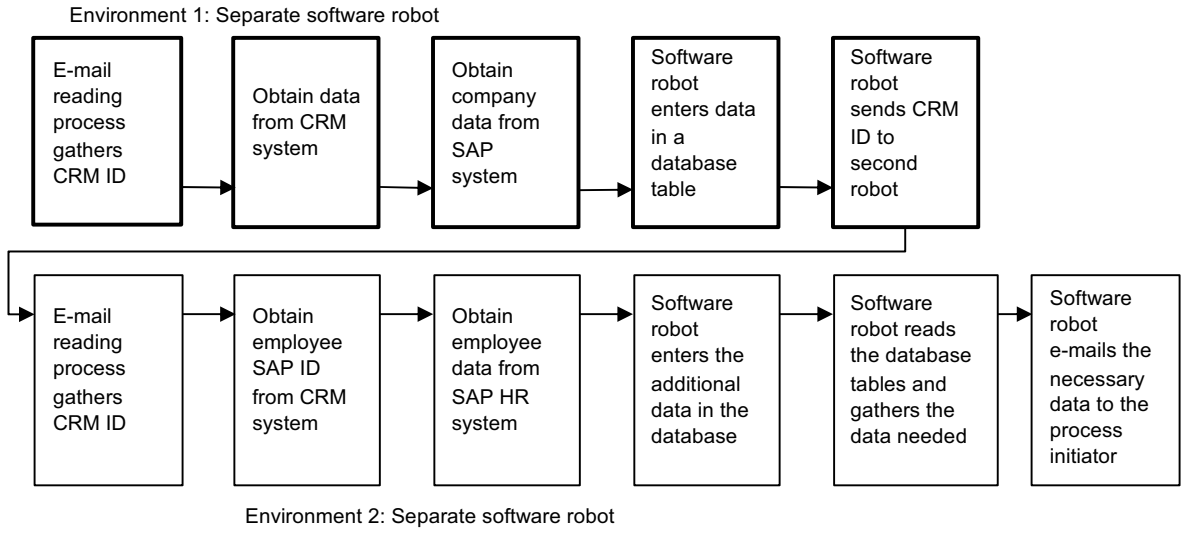

Figure 7. Software robot v. 3 (beta version).

The introduction of a database significantly improved the software-robot speed. Thanks to database operations, the time for processing of each case fell from about 3 min to two, since adding the data retrieved from the database was nearly instantaneous. From Figure 7, one can see the changes newly incorporated into the software-robot system. Processing by the first robot begins in the same way, but in this version it creates a database table, with a name in the format "db+CRM_ID", into which it inserts all the data it has gathered. This naming scheme was used so that the table names would be unique and to afford easy wiping from the database. The second software robot then goes through the same process as before for the first few steps, after which it too adds information to the database. It creates its own table also, named "dbHR+CRM_ID". The reason for creating a separate table for the human-resources data was to keep the process that compiles the data needed from the output of the two software robots easily separable. Once the second robot adds the data that it has gathered to the database, it reads all the data required from both tables and then deletes those tables from the database. After this, it checks the country for which the notification is to be submitted and creates a country-specific email message, which contains all the data needed for a notification to that country. Finally, this email is sent to the person who triggered the software robot.

4.3.3. Reflection on the second BIE cycle

The bulk of the learning from the second BIE cycle stemmed from attempting to resolve the dilemma of secure yet simple RPA design. This would align the efficiency meta-requirement (R3) with the security ones (R1–R2). The efforts to find a win–win solution to the dilemma were driven by security by design's positive-sum principle. Since the additional steps and two-robot structure brought important security improvements, eliminating them was not deemed an option. Efficiency improvements alongside greater security were sought; instead, in the robots' way of handling particular steps and what systems they interact with, not through adjustments in the structure of the artefact itself. The team ended up focusing on the ways of recording data and communicating them between actors engaged in the process. From the naïve perspective of a human interacting with a GUI to perform a process, using Excel files attached to email made intuitive sense; however, the ADR team and end-user team together came to the realisation that RPA could complete the relevant parts of the process more efficiently through creation and destruction of temporary databases. Introducing databases improved the artefact's operational and structural efficiency. Moreover, such a design promised to enhance security, by obviating the need for multiple local copies of data (which would require tracking and, in the final stages, deletion). The latter improvement was aligned well with the security by design approach that guided the research team.

4.4. The third BIE cycle

Developing the third version of the artefact changed how the software robots handled data and responded to external triggers such as email messages. From a process perspective, the software robots' actions now deviated significantly from those performed in handling by humans. The introduction of databases and access controls were adjustments required for ensuring security within a reporting process conducted by a robot. The third BIE cycle was devoted to gathering feedback on the updated artefact and preparing its release version. For this iteration, the primary focus was on improving the robots' reliability and enabling additional assurances that the artefact would adhere to the security standards set by the company.

4.4.1. The feedback on software robot v. 3

The v. 3 form of the artefact too was presented to both the HR worker responsible for the SAP system and the RPA experts. Considerable discussion ensued with regard to what security measures had been taken so far for the handling of sensitive data and what potential problems were still present in the solution. Several needs for minor improvement in the software robots' actions were highlighted, with aims of making error handling and their operation smoother. The main issue pinpointed by the discussion was the need to create some sort of audit trail for the actions of the system. The auditability request arose from the end-user team's need for proof of particular reports' completion in case the receiving countries dispute filing of a report or require evidence of its submission for purposes of their own auditing. In addition, participants stressed that the RPA operators must have the means to track down errors and bugs in the process and eliminate problems with performance and security as soon as possible. Feedback on v. 3 gave impetus to introducing a fourth meta-requirement:

R4: The automation artefact should be transparent and auditable.

4.4.2. Evaluation of results and the building of v. 4

Since the feedback in this cycle focused mainly on the system's auditability, the main issue identified for resolution was making a process trail and technical logging available to its various groups of users, for better security, performance, and compliance. The discussion at the feedback meeting led to the ADR team deciding to articulate an additional meta-requirement. Software robots fulfilling R4 need an audit trail, to enable later cross-checking to verify

that the system has not processed any trips or IDs unnecessarily and for records of who has been using the robots and the process in general. The team determined that at least the CRM ID input and the person who triggered the software robot must be logged.

Accordingly, the practitioner team now had to implement an audit trail for the software robot process. Since the overall process was split into two separate environments, in which two distinct software robots run, the audit process had to cover both of them, to rule out malicious use of either. To track and monitor the use of robotic processing, an additional table was created in the database, for logging the activities of the software robots. Since the practitioner team wanted to track both of the robots, the CRM ID was used for integrated logging data within the database.

Figure 8 includes the two additional steps described above: The first software robot inserts the CRM ID, the email address of the person who sent the trigger email to the robot, and the time at which it was activated into the logging database table. When invoked, the second robot later adds the email address of the entity triggering it (this should always be the address of the first robot) and the time of its invocation to the same table in the database. It uses the CRM ID to identify the correct row for insertion of the data. By affording a later review of who triggered the first robot, what CRM ID was fed in, and when the robots were triggered, the logs create an audit trail sufficient for monitoring the software-robot activities. If specific information is needed on what actual data the robotic processing supplied, the software robots' email folders can be checked (a retention policy was established for the sent-mail folder under which older messages that are no longer relevant get deleted after a certain number of days). The logging data can also be used later on for simple troubleshooting – for instance, if the software-robot process has stopped working, one can see when it was last functioning correctly and pinpoint the CRM ID during whose processing it stopped working. In addition, the log table can reveal whether the first software robot operated successfully but the second did not.

Version 4 of the process was accepted by the parties involved, and the two robots were moved into their

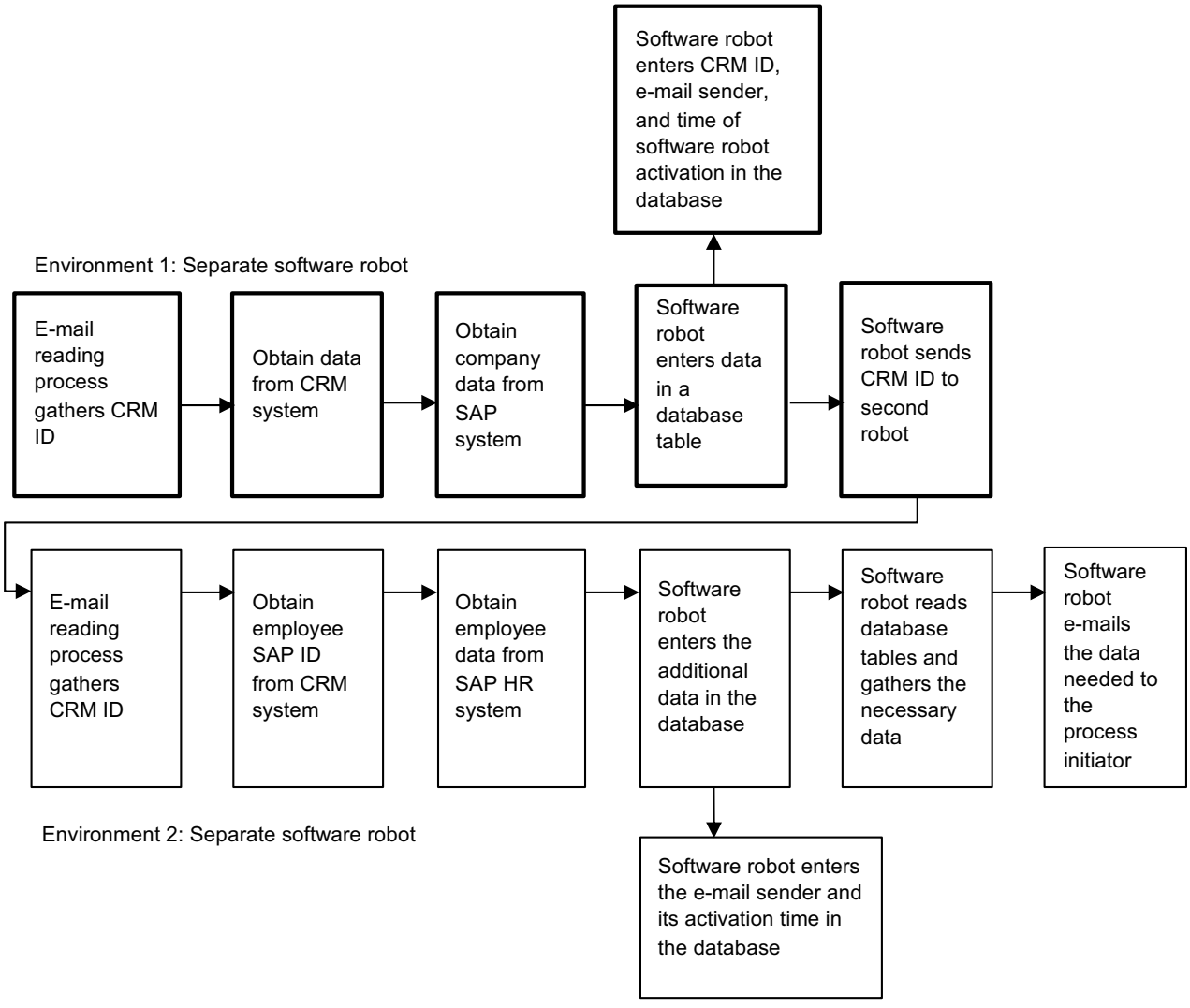

Figure 8. Software robot v. 4 (release 1).

Table 2. The design principles in summary.

Design principle	Meta-requirements addressed	Description
DP1: The Principle of Envelopment	R1, R2, R3	To aid in developers' design of secure software robots, make enterprise architecture design choices that support the creation of envelopes for software robots; they allow the compartmentalisation of mindless software robots and afford enhanced control over the robot, such as removal of temporary-use data.
DP2: The Principle of Access Control	R1, R2	So that developers can design software robots that operate securely, establish practices for the organisation that involve strict control of access to the robotic systems. Such practices enable restricting unauthorised access and ensuring that the robot's actions are confined to only the data needed for the operation.
DP3: The Audit-Trail Principle	R1, R4	To afford developers' design of secure software robots, set up an audit trail to allow tracing and review of the software robot's actions; this trail assigns emulated accountability, a feature necessary for analysis of the thoroughness of the robot's operations and any misuse.

respective virtual environments. The necessary access rights and permissions were set up for the robots, and testing in the production environment began. Then, the process was passed to the control of a new team, established for filing notifications, while the practitioner team supported the technical side of the process and answered questions about EU posted-worker notifications.

4.4.3. Reflection on the third BIE cycle

The final cycle focused on creating a suitable audit trail for the artefact. In this too, security by design's principle of visibility and transparency was very important. The audit-trail functionality extended beyond the specific security concerns identified, and it contributed to technical troubleshooting and compliance. Becoming a crucial component in designing the artefact for security, the audit trail equipped the relevant users with a tool for proactive involvement with the system's security and for identifying issues before they are able to balloon into a major problem. While this feature did add a certain complexity to the artefact design, its impact on process efficiency was negligible.

4.5. Follow-through from the ADR project

To reflect on the findings from our ADR study, we engaged in two types of follow-up. In the first, more than 3 years after deployment of the release version of the software-robot system, we conducted the above-mentioned interviews with two key stakeholders to discuss and reflect on the findings from the study. To a substantial degree, these interviews in December 2021 validated our findings and reflections connected with the BIE interventions, while also shedding light on certain key managerial decisions behind each intervention. The interviews and discussion with the researcher- and practitioner-team members attest that the project can be deemed a success in retrospect, yet three points emerged from the post-implementation reflection as worth further consideration, connected with 1) the complexities associated with the resulting system's legal compliance (specifically, GDPR issues), 2) the maintenance-burden increase arising from a decision to change Wärtsilä's CRM system, and 3) the never-ending need to adjust underlying business processes. Appendix A elaborates on these points and presents a synthesis of the interviews' input.

Secondly, our applicability-check sessions with RPA experts aided in critically assessing the cross-context utility of the design principles, which are discussed in the next portion of the paper. The discussion, centred on the five evaluation criteria adopted from the work of Iivari et al. (2021) (accessibility, importance, novelty, actability, and effectiveness), provided further confirmation of the three principles' utility but also revealed that the RPA practitioners found the envelopment concept new and intriguing. Encouraged by these observations and insight from by the applicability sessions' participants, we ascertained that security by envelopment is a core contribution of our study. Further vital insight emerging from the sessions highlights the layering of the design principles (wherein those pertaining to access control and the audit trail are foundational principles without which the envelopment principle would lack impact). Appendix B presents the associated observations and synthesis of what emerged from the applicability checks.

5. Discussion

At the study's outset, we posed the question of how one can ensure data security in the act of configuring a software robot to automate work processes that involve processing sensitive data. The ensuing ADR study generated three kinds of contributions with specific regard to the class of problem addressed and to the context of software robots operating under constraints imposed by requirements related to the handling of sensitive data. With reference to the core principles of ADR (Sein et al., 2011), we articulate our findings below in terms of these contributions: design principles, theoretical contributions, and practical utility.

5.1. Design principles

Our study generated three design principles (DPs) for configuring security into systems for lightweight automation, such as RPA. To elucidate how these

principles grew out of a design process guided by the theoretical frames of envelopment and security by design and by the four metarequirements, Table 2 maps them to the meta-requirements. We formulated the body text's detailed descriptions of the principles in accordance with the best practice presented by Gregor et al. (2020). Common to all three principles is that they aid in addressing the most fundamental problematic characteristic of software robots: the mindlessness of their operation. The constraint by which they operate from preprogrammed rules and hence cannot cope gracefully with changes to their environment poses special risks in domains wherein the robot must handle sensitive data.

DP1: The Principle of Envelopment:

> To aid in developers' design of secure software robots, make enterprise architecture design choices that support the creation of envelopes for software robots; they allow the compartmentalisation of mindless software robots and afford enhanced control over the robot, such as removal of temporary use data.

The first principle addresses the conflict between the need to grant a mindless automation agent full access to a system containing sensitive data if one desires maximal efficiency (Hofmann et al., 2020) and the best practice of keeping such data secure (Cavoukian & Dixon, 2013). The mindless agent's inability to categorise and compartmentalise distinct types of data and, further, disclose only some of the data, selectively, to parties holding the required access rights poses an immense security risk (Salovaara et al., 2019). While a sophisticated back-end integrated system design might resolve this conundrum, such a solution is not viable for most tasks considered for lightweight automation, where low cost and high efficiency are prioritised. Our alternative, the envelopment principle, emerged from a design solution that implemented proactive security by default through enveloping of automation agents.

To apply the principle of envelopment, the practitioner uses a separate, secure environment as an envelope in which the sensitive-data-gathering part of the automated process is performed. By separating the software operations into multiple environments, one creates envelopes around distinct types of data, with each envelope being accessible only to the relevant parties. Creating mechanisms for more reliable and efficient removal of data from view (e.g., moving over from Excel files to databases) grants the software robot's operator further control over the envelope. In larger organisations especially, numerous automation experts might otherwise have access to the automated processes; hence, it is important to restrict rights to the corresponding environment. In the project, applying this principle not only facilitated more secure automated operation but also made a larger group of RPA experts available to maintain the RPA instance, rather than limiting the support to the few experts permitted to access the closed system containing sensitive data. This contributed further to the system's efficiency and cost-effectiveness.

From the perspective of theory, this principle attests to the envelopment lens's applicability to lightweight-automation agents and provides a foundation for the concept of security by envelopment. We discuss this concept further in the next section on contributions to theory, below.

DP2: The Principle of Access Control:

> So that developers can design software robots that operate securely, establish practices for the organisation that involve strict control of access to the robotic systems. Such practices enable restricting unauthorised access and ensuring that the robot's actions are confined to only the data needed for the operation.

The second design principle is related to access control for triggering a software robot and telling it which information to operate upon. When a robot outputs sensitive data from the systems it uses, there must be appropriate control over who may interact with the robot and how. The principle of access control addresses another limitation associated with a mindlessly operating automaton, related to differentiation among the humans accessing it. Access control also reinforces the envelopment strategy, for secure operations. Configuring separate environments for a system would be ineffectual in the absence of the ability to control access to each environment. Providing clear rules and boundaries for access reduces the number of environmental states that the robot must contend with, thus making it more resilient to unexpected behaviour of external agents.

If wishing to emulate mindful operation, one has to establish a digital mode of human – robot interaction (in our project, email correspondence) and assign unique identifiers to the humans involved (here, each human's company-internal email address). With a limited, well-defined interface of this nature, the robot can identify the human making the request and determine, by following pre-programmed rules, whether to honour that request. Furthermore, associated access rights ought to be limited such that no data other than those truly needed can be accessed or gathered. This design principle for limited access to triggering robotic operations applies specifically to software robots that run in virtual environments ("unattended robots").

DP3: The Audit-Trail Principle:

> To afford developers' design of secure software robots, set up an audit trail to allow tracing and review of the software robot's actions; this trail assigns emulated accountability, a feature necessary for analysis of the thoroughness of the robot's operations and any misuse.

The third design principle emerged in response to the requirement for transparency and auditability (R4). Such a requirement is aligned with the best practice of security by design, wherein proactivity is combined with visibility and transparency (see Cavoukian & Dixon, 2013). With this principle, we posit that one should create an audit trail for the process. If the above-mentioned controls on the use of the automated process do not fully guarantee its use for the intended purpose and/or it is applied for malicious ends, this ensures that those making use of the process can be tracked in practice. Also, in the event of a systems audit, whether internal or external, there needs to exist proof of how the process has been utilised and when.

This design principle aids in assigning emulated accountability for the actions of an automation agent. As software robots operate mindlessly, they are unable to apply any judgement as to the correctness of their actions. Therefore, a software-robot system designed with security in mind should be endowed with an ability to document its actions and to present the documentation in question to human operators in a readily understandable and actionable form. Applying the principle of guaranteeing an audit trail reinforces the other two design principles by ensuring their effective fulfilment and providing input for further configuration improvements in pursuit of security.

5.2. Contributions to theory

Our study enriches scholarly discussion of configuring lightweight-automation systems while taking heed of issues pertaining to organisational data security. Guided by recent theoretical developments related to mindless digital processing (Salovaara et al., 2019), sociotechnical envelopment (Asatiani et al., 2021; Robbins, 2020), and the security by design approach (Cavoukian & Dixon, 2013), our ADR study enabled us to derive three vital principles for designing a secure lightweight-automation artefact. While the lightweight nature of the system resulted in the design output remaining cost-effective and not impinging on the case company's existing IT architecture, the artefact was found to display high security nonetheless. How did the organisation reach this balance? The answer is a new theoretical concept that emerged as fruit of our study: security by envelopment, or SbE. It allows embedding some of the principles previously identified as crucial for security (such as transparency and strong emphasis on the protection of user data, which are central to security that is proactive by default) but without prohibitive effort. Thus, SbE addresses the necessary balance between efficiency and security in the development of lightweight-automation systems that handle sensitive data.

Many traditional approaches to rendering and keeping information systems secure call for their redesign such that security forms a core element of the system design. Applying such techniques can lead to reduced cost-effectiveness and lower efficiency, so automation weighed down by them is seldom deemed viable for low-business-value administrative tasks. Although such tasks, often considered purely cost-generating, are perfect candidates for automation, their non-core nature simultaneously leaves organisations unwilling to invest significantly in automating them carefully. In contrast, SbE offers a way of solving this problem by introducing the concept of envelopment surrounding the lightweight-automation system. Our new solution permits retaining the original design of the lightweight automation. A developer adhering to SbE aims to create a secure envelope around the artefact, to ensure data security and thereby protect it.

The envelopment proactively makes the system secure by controlling access, function range, inputs, and outputs, all while maintaining the system's efficiency in development and use. The core idea behind SbE is that containing potential security risks within an envelope around the artefact makes the organisation able to retain the efficiency represented by lightweight automated systems and safely open more opportunities for their use, in a wider range of contexts. SbE follows both the principles of sociotechnical envelopment (Asatiani et al., 2021) and security by design (Cavoukian, 2009) in considering the social component of IS configuration. This manifests in both tweaking the underlying task to fit software robots' properties and minimising exposure of humans interacting with the robot to sensitive data, as compared to when humans performed the same process manually. The SbE approach speaks to the increasing recognition of data security's importance in organisational environments that feature quasi-intelligent automation agents (e.g., Bygstad, 2017; Moffitt et al., 2018).

5.3. Practical utility

The RPA project and the new automated process generated significant practical utility for the organisation and its key stakeholders. In the organisational context of our study, the practical challenge was to automate the postedworker notifications while adhering to the ever-tightening GDPR requirements. Practical utility essentially emerged from the interventions made in the project that facilitated the balancing act of taking into account the requirements for process efficiency on one hand and ensuring the security of data on the other hand.

For practical utility from the *efficiency perspective*, the organisation was able to deploy the robot to automate the heavy manual reporting process associated with making the posted-worker notifications. It was found that, for example, by retrieving and e-mailing back only the information required for the notification in question, the

artefact reduces the time needed for finding and compiling the details by 10–30 min per notification for the end user, according to the organisation's estimates. In a setting such as this, with thousands of international assignments each year, these time savings represent substantial cumulative benefits. A by-product of the interventions conducted during the ADR study pertains to a more precise process documentation related to the notifications. These improvements in the level of detail in process documentation lay fruitful ground for process efficiency enhancements in the future.

For practical utility from the *security perspective*, the three design principles outlined above demonstrate the essence of how the organisation was able to design an artefact of this nature in such a manner that security is ensured in handling sensitive data. Collectively, these design principles provide a framework for building solid automated processes in the future work and constitute a proactive stance to design for an organisation of this sort: adhering to them provides the practical utility of not having to worry about abuse of software robots or about reactively trying to fix the robot further down the line. The principle associated with the removal of temporary-use data (i.e., ensuring that old copies of Excel files or data residing in databases are duly deleted before completing the process) remains as a key part of the organisation's RPA development to this day. Another practice that the organisation has retained for all their automation projects is related to access control; human-triggered automation is no longer email triggered, but instead, the organisation has created a digital platform where software robots can be launched, providing enhanced transparency to human-triggered automation.

5.4. Limitations and further research

While we are confident in our design approach and report on a study with quality, richness, volume, and validity of data, we acknowledge the limitations related to the ADR project's scope and to the transferability of the design principles presented here to similar problem classes. We should stress, firstly, that we dealt with only the implementation stage in this project. Full deployment and maintenance of the software robot designed were beyond the scope of our work, and the study concluded by examining the design principles associated with its iterative creation process. We collected follow-up data on the overall success of the project, 2 years after its completion; however, no systematic account of deployment and maintenance was included. Extending the time horizon by following the robot from deployment through to long-term maintenance could yield additional design principles. Research taking a longitudinal approach could examine evolution and our results' continued relevance on the larger stage too, as the process matures in the light of companies' cumulative experience and results such as ours. Secondly, considering generalisation of our design principles to problem classes of related types, we remind the reader that we studied configuring an information systems artefact to handle sensitive data with specific regard to a category of software robots that can be regarded as an instantiation of lightweight-automation tools. Accordingly, some traits specific to software robots and the particulars of our project (e.g., triggering by email messages) may have influenced the design principles put forth in this paper. Therefore, the concept of SbE is focused only on the security of sensitive data, as our data does not encompass other facets of information systems' security. It would be valuable for further research to study whether similar design principles may be employed in contexts of configuring more advanced, back-end automation tools and whether these principles may solve a broader set of security problems.

6. Conclusion

With this paper, we have proposed a novel concept – security by envelopment – that facilitates secure operation of lightweight-automation systems without undermining their efficiency and cost-effectiveness. This approach attends to the pervasive need for designing RPA processes carefully to meet requirements aligned with a rapidly changing regulatory landscape (e.g., such that organisations comply with data-protection rules). Our ADR-based approach to designing an artefact for automation of a specific reporting process that involves processing employees' sensitive data proved fruitful not only for producing a software robot solution for handling EU posted-worker notifications but also for advancing research into the design of organisational data-security practices, particularly in the realm of lightweight automation.

Notes

1. The Posted Workers Directive (European Commission, n.d..), under which companies must notify EU authorities about all posted workers, specifies a set of mandatory rules for the terms and conditions of employment to be applied for a posted worker (an employee sent by the employer to perform a service in another Member State on a temporary basis). The aim behind the directive is to safeguard the rights of posted workers and appropriate work conditions, ensure a level playing field, and avoid "social dumping" whereby foreign service providers can undercut local ones by following less strict labour standards.
2. The pioneering work to create RPA responsive to the growing demands that new regulatory instruments impose on routine work is consistent with Wärtsilä's broader objective of decentralising RPA development and bringing it closer to the base of the organisational pyramid. One example of these efforts is the participation of 900-plus workers in Wärtsilä's RPA training

sessions, with more than 100 "citizen developers" assisting in the RPA's further development.
3. We thank an anonymous reviewer for suggesting applicability-check sessions.
4. This stage also included gathering information on country-specific transposition of the directive, by consulting each destination country's Web site dedicated to matters of posted workers. Because each of the countries has had some leeway to interpret it in a manner favourable to it while following the guidelines from the EU, the implementation model for the notification obligation differed slightly from country to country, ranging from sending the notification by post to utilising various types of online forms and portals.
5. UiPath is a popular software suite for RPA. It allows users to program automation via a graphical drag-and-drop interface, by using pre-built activities, or through a programming interface for the Visual Basic and C# programming languages, to build customised operations.

Acknowledgements

We wish to thank the Senior Editor, Associate Editor, and three anonymous reviewers for their invaluable contributions to this paper. Their insightful comments and constructive feedback greatly improved the quality of our research. Furthermore, we extend our appreciation to Wärtsilä Corporation for participating in the study and generously providing us with access to the necessary data. We would also like to acknowledge the participants of the Global Sourcing Workshop 2019 (Obergurgl, Austria) and the participants of research seminars held at the University of Queensland (Australia) and the University of Gothenburg (Sweden) for their valuable feedback. Lastly, we would like to thank Professor Johan Magnusson for his detailed feedback on an early version of our paper.

Disclosure statement

One of the authors, Kimmo Paaso, is currently employed at the Wärtsilä Corporation.

ORCID

Aleksandre Asatiani http://orcid.org/0000-0002-7358-9018
Tuuli Hakkarainen http://orcid.org/0000-0002-3144-9236
Esko Penttinen http://orcid.org/0000-0003-0316-7538

References

Allianz Global Corporate & Speciality. (2019). Allianz risk barometer: top business risks for 2019. *Allianz Risk Pulse*, 12–15. https://www.agcs.allianz.com/content/dam/onemarketing/agcs/agcs/reports/Allianz-Risk-Barometer-2019.pdf

Anderson, C., Baskerville, R. L., & Kaul, M. (2017). Information security control theory: achieving a sustainable reconciliation between sharing and protecting the privacy of information. *Journal of Management Information Systems*, *34*(4), 1082–1112. https://doi.org/10.1080/07421222.2017.1394063

Angst, C. M., Block, E. S., D'arcy, J., & Kelley, K. (2017). When do IT security investments matter? Accounting for the influence of institutional factors in the context of healthcare data breaches. *MIS Quarterly, 41*,(3), 893–A8. 24 January .

Asatiani, A., Malo, P., Nagbøl, P. R., Penttinen, E., Rinta-Kahila, T., & Salovaara, A. (2020). Challenges of explaining the behavior of black-box ai systems. *MIS Quarterly Executive*, *19*(4), 259–278. https://doi.org/10.17705/2msqe.00037

Asatiani, A., Malo, P., Nagbol, P. R., Penttinen, E., Rinta-Kahila, T., Salovaara, A., Nagbøl, P. R., Penttinen, E., Rinta-Kahila, T., & Salovaara, A. (2021). Sociotechnical envelopment of artificial intelligence: an approach to organizational deployment of inscrutable artificial intelligence systems. *Journal of the Association for Information Systems*, *22*(2), 325–352. https://doi.org/10.17705/1jais.00664

Asatiani, A., & Penttinen, E. (2016). Turning robotic process automation into commercial success - Case OpusCapita. *Journal of Information Technology Teaching Cases*, *6*(2), 1–8. https://doi.org/10.1057/jittc.2016.5

Bygstad, B. (2017). generative innovation: a comparison of lightweight and heavyweight IT. *Journal of Information Technology*, *32*(2), 180–193. https://doi.org/10.1057/jit.2016.15

Cavoukian, A. (2009). *Privacy by Design The 7 Foundational Principles Implementation and Mapping of Fair Information Practices*. September 01, 2022. https://privacy.ucsc.edu/resources/privacy-by-design---undational-principles.pdf

Cavoukian, A., & Dixon, M. (2013). *Privacy and security by design: An enterprise architecture approach*. Information and Privacy Commissioner of Ontario.

Chen, P., Kataria, G., & Krishnan, R. (2011). Correlated failures, diversification, and information security risk management. *MIS Quarterly*, 397–422. https://doi.org/10.2307/23044049

Clancey, W. J. (1985). Heuristic classification. *Artificial Intelligence*, *27*(3), 289–350. https://doi.org/10.1016/0004-3702(85)90016-5

Cram, W. A., Proudfoot, J. G., & D'arcy, J. (2017). Organizational information security policies: A review and research framework. *European Journal of Information Systems*, *26*(6), 605–641. https://doi.org/10.1057/s41303-017-0059-9

Culnan, M. J. (2000). Protecting privacy online: Is self-regulation working? *Journal of Public Policy & Marketing*, *19*(1), 20–26. https://doi.org/10.1509/jppm.19.1.20.16944

Culnan, M. J., & Williams, C. C. (2009). How ethics can enhance organizational privacy: Lessons from the choicepoint and TJX data breaches. *MIS Quarterly*, *33*(4), 673–687. https://doi.org/10.2307/20650322

D'arcy, J., & Herath, T. (2011). A review and analysis of deterrence theory in the is security literature: Making sense of the disparate findings. *European Journal of Information Systems*, *20*(6), 643–658. https://doi.org/10.1057/ejis.2011.23

DiMaggio, P. J., & Powell, W. W. (1983). The iron cage revisited: institutional isomorphism and collective rationality in organizational fields. *American Sociological Review*, *48*(2), 147–160. https://doi.org/10.2307/2095101

Earp, J. B., Antón, A. I., Aiman-Smith, L., & Stufflebeam, W. H. (2005). Examining Internet privacy policies within the context of user privacy values. *IEEE*

Transactions on Engineering Management, *52*(2), 227–237. https://doi.org/10.1109/TEM.2005.844927

European Commission. (n.d). *Posted workers*. Employment, Social Affairs & Inclusion. Retrieved April 3, 2020, from https://ec.europa.eu/social/main.jsp?catId=471

European Union. (2016). Regulation (EU) 2016/679 of the European parliament and the council. *Official Journal of the European Union*, *119*(1), 1–88. https://doi.org/10.5771/9783845266190-974

Floridi, L. (2011). Children of the fourth revolution. *Philosophy & Technology*, *24*(3), 227–232. https://doi.org/10.1007/s13347-011-0042-7

Greenaway, K. E., & Chan, Y. E. (2005). Theoretical explanations for firms' information privacy behaviors. *Journal of the Association for Information Systems*, *6*(6), 171–198. https://doi.org/10.17705/1jais.00068

Greenaway, K. E., Chan, Y. E., & Crossler, R. E. (2015). Company information privacy orientation: A conceptual framework. *Information Systems Journal*, *25*(6), 579–606. https://doi.org/10.1111/isj.12080

Gregor, S., Chandra Kruse, L., & Seidel, S. (2020). Research perspectives: The anatomy of a design principle. *Journal of the Association for Information Systems*, *21*(6), 1622–1652. https://doi.org/10.17705/1jais.00649

Haag, S. (2015). *Appearance of dark clouds? – an empirical analysis of users ' shadow sourcing of cloud services*. Wirtschaftsinformatik, Osnabrück, Germany. 1438–1452.

Haj-Bolouri, A., Purao, S., Rossi, M., & Bernhardsson, L. (2018). *Action design research in practice: Lessons and concerns*. Proceedings of the 26th European Conference on Information Systems, Portsmouth, UK. 1–16.

Hallikainen, P., Bekkhus, R., & Pan, S. L. (2018). How OpusCapita used internal rpa capabilities to offer services to clients. *MIS Quarterly Executive*, *17*(1), 41–52.

Hofmann, P., Samp, C., & Urbach, N. (2020). Robotic process automation. *Electronic Markets*, *30*(1), 99–106. https://doi.org/10.1007/s12525-019-00365-8

Iivari, J., Rotvit Perlt Hansen, M., & Haj-Bolouri, A. (2021). A proposal for minimum reusability evaluation of design principles. *European Journal of Information Systems*, *30*(3), 286–303. https://doi.org/10.1080/0960085X.2020.1793697

Kwon, J., & Johnson, M. E. (2014). Proactive versus reactive security investments in the healthcare sector. *MIS Quarterly*, *38*(2), 451–471. https://doi.org/10.25300/MISQ/2014/38.2.06

Lacity, M., & Willcocks, L. (2016). Robotic Process Automation at Telefónica O2. *MIS Quarterly Executive*, *15*(1), 21–35.

Langheinrich, M. (2001). Privacy by design - Principles of privacy-aware ubiquitous systems. *Lecture Notes in Computer Science*. https://doi.org/10.1007/3-540-45427-6_23

Lin, C., Wittmer, J. L. S., & Luo, X. R. (2022). Cultivating proactive information security behavior and individual creativity: The role of human relations culture and IT use governance. *Information\& Management*, *59*(6), 103650. https://doi.org/10.1016/j.im.2022.103650

Lowry, P. B., Dinev, T., & Willison, R. (2017). Why security and privacy research lies at the centre of the information systems (IS) artefact: Proposing a bold research agenda. *European Journal of Information Systems*, *26*(6), 546–563. https://doi.org/10.1057/s41303-017-0066-x

Mehrizi, M. R., Nicolini, D., & Modol, J. R. (2022). How do organizations learn from information system incidents? a synthesis of the past, present, and future. *Management Information Systems Quarterly*, *46*(1), 531–590. https://doi.org/10.25300/MISQ/2022/14305

Mendling, J., Decker, G., Germany, S., Hull, R., & Reijers, H. A. (2018). How do machine learning, robotic process automation, and blockchains affect the human factor in business process management? *Communications of AIS*, *43*(June), 1–23. https://doi.org/10.17705/1CAIS.04319

Moffitt, K. C., Rozario, A. M., & Vasarhelyi, M. A. (2018). Robotic process automation for auditing. *Journal of Emerging Technologies in Accounting*, *15*(1), 1–10. https://doi.org/10.2308/jeta-10589

Mullarkey, M. T., & Hevner, A. R. (2019). An elaborated action design research process model. *European Journal of Information Systems*, *28*(1), 6–20. https://doi.org/10.1080/0960085X.2018.1451811

Penttinen, E., Kasslin, H., & Asatiani, A. (2018). *How to choose between robotic process automation and back-end system automation?*. European Conference on Information Systems 2018, Portsmouth, UK. 1–14.

Posey, C., Roberts, T., Lowry, P., Bennett, R., & Courtney, J. (2013). Insiders' protection of organizational information assets: Development of a systematics-based taxonomy and theory of diversity for protection-motivated behaviors. *MIS Quarterly*, *37*(4), 1189–1210. https://doi.org/10.25300/MISQ/2013/37.4.09

Purao, S., Henfridsson, O., Rossi, M., & Sein, M. (2013). Ensemble artifacts: from viewing to designing in action design research. *Systems, Signs & Actions*, *7*(1), 73–81.

Robbins, S. (2020). AI and the path to envelopment: Knowledge as a first step towards the responsible regulation and use of AI-powered machines. *AI & Society*, *35*(2), 391–400. https://doi.org/10.1007/s00146-019-00891-1

Rosemann, M., & Vessey, I. (2008). Toward improving the relevance of information systems research to practice: the role of applicability checks. *MIS Quarterly*, *32*(1), 1–22. https://doi.org/10.2307/25148826

Salovaara, A., Lyytinen, K., & Penttinen, E. (2019). High reliability in digital organizing: mindlessness, the frame problem, and digital operations. *MIS Quarterly*, *43*(2), 555–578. https://doi.org/10.25300/MISQ/2019/14577

Sarker, S., Chatterjee, S., Xiao, X., & Elbanna, A. (2019). The sociotechnical axis of cohesion for the is discipline: Its historical legacy and its continued relevance. *MIS Quarterly: Management Information Systems*, *43*(3), 695–719. https://doi.org/10.25300/MISQ/2019/13747

Scheel, P. D. (1993). Robotics in industry: A safety and health perspective. *Professional Safety*, *38*(3), 28.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. I. (2011). Action Design Research. *MIS Quarterly*, *35*(1), 37–56. https://doi.org/10.2307/23043488

Sein, M. K., & Rossi, M. (2019). Elaborating ADR while drifting away from its essence: A commentary on Mullarkey and Hevner. *European Journal of Information Systems*, *28*(1), 21–25. https://doi.org/10.1080/0960085X.2018.1527189

Stallings, W. (2020). Handling of personal information and deidentified, aggregated, and pseudonymized information under the California consumer privacy act. *IEEE Security\& Privacy*, *18*(1), 61–64. https://doi.org/10.1109/MSEC.2019.2953324

Stople, A., Steinsund, H., Iden, J., & Bygstad, B. (2017). Lightweight IT and the IT function: experiences from robotic process automation in a Norwegian bank. *Bibsys Open Journal Systems*, *25*(1), 1–11.

Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J. J., Ouyang, C., Ter Hofstede, A. H. M., van de Weerd, I., Wynn, M. T., & Reijers, H. A. (2020). Robotic process automation: contemporary themes and challenges. *Computers in Industry*, *115*, 115. https://doi.org/10.1016/j.compind.2019.103162

Symantec Corporation. (2019). Internet security threat report. *Network Security*, *24*(February).

UiPath. (n.d.). Wärtsilä: 400 automated processes supported by a citizen developer community. Accessed 01 September, 2022. https://www.uipath.com/resources/automation-case-studies/wartsila-marine-energy-market-rpa

Vance, A., Lowry, P. B., & Eggett, D. (2015). Increasing accountability through user-interface design artifacts. *MIS Quarterly*, *39*(2), 345–366. https://doi.org/10.25300/MISQ/2015/39.2.04

van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. *Business & Information Systems Engineering*, *60*(4), 269–272. https://doi.org/10.1007/s12599-018-0542-4

Willcocks, L., Lacity, M., & Craig, A. (2015). *The IT Function and Robotic Process Automation* London, U: The Outsourcing Unit Working Research Paper Series.

Appendix A: Follow-up interviews with key stakeholders at Wärtsilä

To delve more deeply into the findings from our ADR study, we conducted interviews with two key stakeholders from Wärtsilä (a line manager who was in charge of the Posted Worker project and a project manager). The purpose of these interviews, conducted in December 2021, was to obtain retrospective reflective feedback on how the project went and evaluate the impact of the final software robot after its deployment.

Firstly, ensuring the legal compliance of the resulting system turned out to be more complex than anticipated. The personal data connected with EU trips were copied and stored in multiple locations for purposes of providing input to the dashboard and affording auditing of the RPA agents' actions. While the artefact adhered to the principle of data minimisation, neither the dashboard nor the audit-trail features could function without temporary copies of the data. This necessitated extensive dialogue with the legal experts at Wärtsilä. There was a lack of established best practice with regard to the GDPR, since this regulation was fairly new, accompanied by uncertainty surrounding various specifics of implementing and designing GDPR-compliant systems. Moreover, the project marked the company's first experience with a software robot handling sensitive HR details and delivering relevant pieces of said data to an employee who lacks direct access rights for the data. Therefore, the legal experts had to carefully consider the implications of potential for the robot's unintended leaking of data to an unauthorised employee. Approval of the artefact came eventually, once the practitioner team had delivered assurances that its design renders such accidents impossible.

The second item identified for reflection is related to updates made to the CRM system's user interface. One of the main factors prompting RPA system updates is changes in the user interface of systems with which a robot interacts (Stople et al., 2017). Because RPA agents are programmed to interact via user interfaces, even minor changes in a user-facing interface could "break" RPA operations. This issue manifested itself when Wärtsilä decided to update its CRM system after the RPA deployment: that update included user-interface changes that led to the artefact no longer working. The RPA needed reprogramming in line with the new interface elements. While aligning the RPA code with a new interface seldom requires much time, the CRM update disrupted the process of filing EU posted-worker notification and did demand RPA resources. The incident concretised a need for maintainers of various IT systems to co-ordinate their updates with those managing the RPA, so as to prevent interruptions to vital operations. Doing so may not always be easy, however, especially in cases involving cloud-based systems provided and maintained by a third party.

Thirdly, stakeholders involved in the process have learnt that balance must be maintained between technology and process development. At its core, RPA is a fairly simple technology that can be developed and deployed within mere days. However, in practice, developing a well-functioning RPA implementation that fulfils all business requirements and accounts for all the relevant constraints may take quite some time. In our case, the organisation needed to rethink the process of collecting the data from the various systems, expedite that process, and honour-specific legal restrictions imposed by the GDPR. Accordingly, developing the artefact took 6 months. Since Wärtsilä had amassed experience of neither the EU posted-worker notification process's development nor the GDPR's requirements, the exercise required several iterations before the artefact satisfied the needs of all stakeholders. There is an important lesson here for organisations hoping to embark on such an RPA journey. While RPA is advertised as a cost-efficient and easily implemented technology, it does not absolve one of the needs for thorough process development. Since that development could end up demanding much more time and money than expected, organisations' cost–benefit analysis should take this into account.

Appendix B: Applicability checks

For a critical evaluation of the generalisability of the design principles developed through the project, we conducted applicability checks with practitioners (see Iivari et al., 2021; Rosemann & Vessey, 2008). These were carried out in December 2022, a full three years] after deployment of the release version of the artefact. This appendix presents the related preparations, protocol, and analysis.

Development and preparation for the applicability-check sessions

A brief summary of the key insight from the design study was developed prior to embarking on the applicability checks. This four-page summary included a description of the practical problem faced by the company, followed by a recapping of each design principle (see the manuscript's "Discussion" section). The summary, which also contained figures presenting the initial and final versions of the RPA (Figures 3 and 7 in the manuscript, respectively), was sent to the informants beforehand, and they were asked to read it. No other preparation was requested of them.

The sampling for the applicability checks chose informants who would be knowledgeable with regard to RPA and possess extensive experience in configuring it for various contexts. Three sessions were held: a focus-group session with two informants and two single-informant sessions (see Table B1 below). Each session lasted one hour and was audio-recorded, with later transcription.

Table B1. The applicability sessions.

Session	Informants	Format
Session 1	Solutions consultant Technology lead	Face-to-face
Session 2	Head of business development	Virtual, via Zoom
Session 3	Director of automation	Face-to-face

The applicability-check protocol and analysis

Each session started with probing initial reactions to the material that had been sent prior to the session. Then, the informants were asked for their insight related to each of the three design principles. Discussion of the concept of sensitive data followed. Finally, we asked specific questions covering the five evaluation criteria: accessibility, importance, novelty and insightfulness, actability and appropriate guidance, and effectiveness (see Iivari et al., 2021).

What were your initial reactions to the short summary?

What does "sensitive data" mean to you? Have you participated in projects wherein RPA would be configured to handle potentially sensitive personal data? If so, how did your approach to those projects differ from some other projects'?

What did you think of the first design principle, envelopment?

What did you think of the second design principle, related to access control?

What did you think of the third design principle, for audit trails?

Accessibility

The design principles are easy for me to understand

The design principles are intelligible to me

Importance

In my view, this RPA project addresses a real problem that could be generalised to my professional practice

In my view, this RPA project addresses an important – acute or foreseeable – problem in my professional practice

Novelty and insightfulness

I find that the design principles present me with new ideas

I find that the design principles offer insight for my own practice

Actability and appropriate guidance

I think the design principles can realistically be acted upon in practice

I find that the design principles provide sufficient guidance for designing secure RPA systems

I find that the design principles are not restricted to designing secure RPA systems

I find that the design principles provide me with sufficient freedom for designing secure RPA systems

Effectiveness

Effectiveness in my organisation:

I believe that the design principles can aid in designing secure RPA systems

I find the design principles potentially useful for designing secure RPA systems in practice

I believe that secure RPA systems would improve the quality of the products/services of my organisation relative to the current situation

Effectiveness in my own work:

I believe that secure RPA systems would improve my performance in comparison to my current situation

Together, the three sessions yielded 14,387 words of transcribed text. That text was analysed by means of coding using the software ATLAS.ti. We began our analysis by coding the informants' initial reactions to the brief summary that was sent to the informants beforehand, then proceeded to coding the informants' insights to each of the three design principles. We also coded each informant's views on the five evaluation criteria. The remainder of the coding process was inductive and iterative in nature and yielded additional codes related to definitions of sensitive data, layered nature of the design principles, and RPA user credentials. Collectively, the coding process has resulted in the creation of 14 open codes, with 46 quotes. The coding scheme along with illustrative codes is available from the authors upon request.

Insight from the applicability checks and implications related to the main study

When contemplating the criteria of accessibility, importance, novelty, actability, and effectiveness, the informants stated that the term "envelopment" was new to them and interesting. This did mean that, to grasp its essence in the context of secure software robots, the informants had to read the description of the associated design principle carefully. The other two principles (related to access control and audit trails) were more familiar to them and considered less novel. All informants felt that concerns surrounding sensitive data and security issues are important, classic problems associated with software robots in a practical business context. The informants did not see any limitations to bringing these design principles into use, though some did note that runtime issues might become an issue as the structure of the robot grows more complex. The discussion about the evaluation criteria gave the researcher team confidence that the design principles derived via the project possess value.

In the applicability-check sessions, we talked at length about hazardous work-task combinations. It is common practice to have separate human workers handle discrete parts of a process (one informant cited the classic example of processing invoices; the same person should not be approving the invoice and having authority to modify account information). The informants felt that the use of envelopment and multiple robots for different parts of the process could be a mechanism for avoiding risks associated with hazardous task combinations in robotic processes.

Further to this discussion, one informant pointed out another potential critical security concern in robotised processes: a human can use the robot's credentials to perform illegal actions in the systems. The informants found that combining the first two design principles (related to envelopment and access rights) could be an interesting approach to dealing with such risks. These observations indicated to the researchers that the design principles are interrelated and that their impacts on security are contingent upon each other.

Regarding the interaction among the design principles, several discussions revolved around the idea of layering of the principles. The envelopment principle stood out among the three as an impactful principle for tackling security concerns, while the other two were regarded as foundations – necessary conditions for any impact from applying the envelopment principle. This suggests a hierarchy of fundamental and "means" principles, with access rights and audit trails being among the former while envelopment is one of the latter.