Case Study Protocol

Markus Funke
Vrije Universiteit Amsterdam

Working Title

A Process for Monitoring the Impact of Architecture Principles on Sustainability

Preamble

This case study protocol intends to capture and describe the research design together with all taken design decisions. To ensure a consistent planning process, this document follows the protocol template of Brereton et al. (2008) and the case study research guidelines of Runeson and Höst (2008). Since a case study methodology follows a flexible design strategy, this document is a living document in its nature and is reviewed, adjusted and edited at each iteration of the case study process.

Background

Problem

Software architectural principles in software systems are usually created on the strategic or tactical level and embedded in the software's architecture. Current research and the industry tends to discuss principles and their impact on only certain aspects of sustainability—or a combination of them (e.g., social and technical, or, economic and technical). However, to achieve a sustainable software system in the first place, architectural principles should be selected concerning all four sustainability dimensions: i) technical, ii) economic, iii) environmental, and iv) the social dimension. Novel research has proposed such methods to address sustainability for software systems—but rather during the planning phase or to map already existing solutions on the different dimensions.

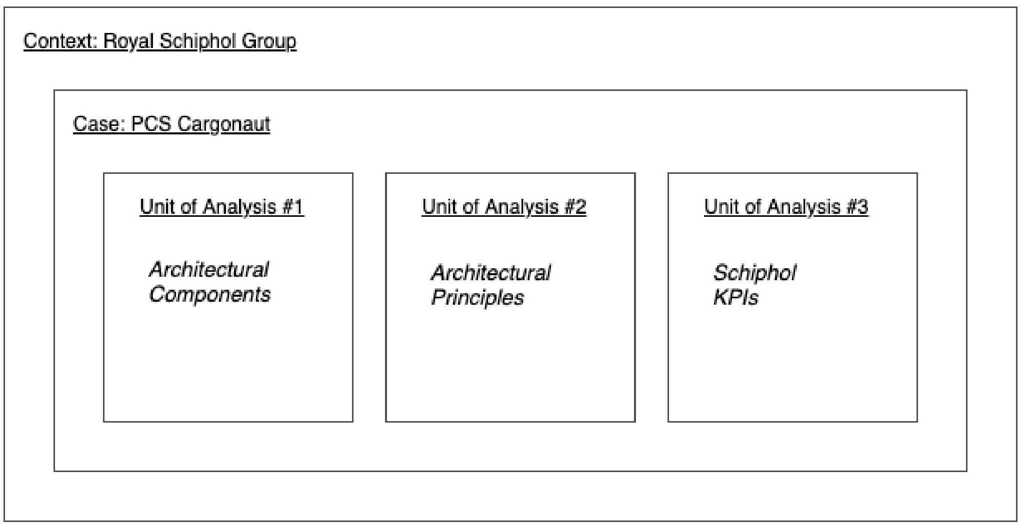
To understand the long-term extension of software architecture principles to sustainability, it is necessary to monitor the impact over time. Long-term observations would help to identify the positive or negative effects of architectural principles on sustainability and can then support the architectural process on the strategic or tactical level. To achieve such monitoring, it is necessary to quantify and monitor the impact of architectural principles on every sustainability dimension.

General Procedures

Our case study follows the five major process steps for a 'case study research process' according to Runeson and Höst (2008):

1) Case study design:
   objectives are defined and the case study is planned.
2) Preparation for data collection:
   procedures and protocols for data collection are defined.
3) Collecting evidence:
   execution with data collection on the studied case.
4) Analysis of collected data
5) Reporting

Design



This study is considered as a holistic case study in one company and with one certain project as a research case. Although, it might be seen as an embedded case study, as we do have multiple units of analysis. We conduct the research in the **context** of the *Royal Schiphol Group (RSG)*. The RSG was selected based on existing academia-industry relations between the Vrije Universiteit Amsterdam and CGI Nederland. The RSG provides the **case** *Port Community System (PCS) - Cargonaut*. We identified three different **units of analysis** to answer our research questions, namely the (i) architectural components, (ii) architectural principles, and the (iii) Schiphol KPIs.

Theory

Concerning the frame of reference, no explicit theories are referred to in this study. However, the investigated approaches are based on existing methods that, to some extent, already have been investigated. Earlier studies thereby affect the design of this study. This case study is partly a replication of the work from Gupta et al. (2021). The authors propose a framework to map architectural principles on the four sustainability dimensions. They used their own OGSM model, namely PRSM model (Principle, Rationale, Strategies, Measures), and Decision Maps from Lago (2019) to achieve the mapping of principles to sustainability. However, the results of Gupta et al. (2021) can only be partially used due to a redefined IT strategy at RSG. In addition, the PCS - Cargonaut uses different and newly defined architecture principles.

Research Instruments

This section provides the study design and research instruments at hand to describe the case study process.

Case Selection

The case is selected based on various solutions provided by RSG. Each solution is analyzed in depth and according to different criteria:

- **Availability**: A key criterion for the case selection is the availability of the case itself. Access through documents, experts, or direct access is indispensable. The research results depend closely on the access to the case and the availability of the case data such that data can be collected unrestrictedly.
- **Confidentiality**: The case must not be subject to secrecy. To conduct the research and especially publish the research results, the case and the related data need to be accessible by the researcher. However, since a case study is often related with confidential information of an organization, we consider the key ethical factors from Runeson and Höst such as Informed consent, Confidentiality, and Handling of sensitive results.
- **Case volume**: The case volume must be large enough to obtain sufficient data and the time required to obtain the information must be appropriate to the scope of the investigation.
- **Development status**: For the case selection, it is not critical whether the solution is already in production, still under development, or has the status Proof of Concept (PoC). However, this criterion is considered along with the question to what extend data can be extracted and examined. The decision is taken from case to case.
- **Relevance**: This research is interested in the software architectural principles and their impact on sustainability. Hence, a solution needs to be based and developed upon architectural principles. In addition, the software solution, i.e. the case un- der research should be relevant to contribute towards the company's strategy and especially the company's sustainability strategy.
- **Completeness**: Instead of focusing on a case that covers only one particular group of software architectural principles, the solution under research should be selected to address a diverse range of principles. This would help to gather as much data as possible, and to derive conclusions beyond the specific case. In addition, different architectural principles help to identify also a broad range of company-wide KPIs and their impact on sustainability. Conclusions could be applied to other company-wide solutions and beyond company borders.

The following solutions are provided by RSG:

- RPA
- ASB2.0 (Airport Service Bus)
- ETO building blocks
- MFT
- CEP (Complex Event Processing
- **PCS Cargonaut (Port Community System) → chosen → see final report for rationales and answers to the prior defined characteristics.**

Based on the solution evaluation below, the PCS case evolved as best fit for our research:

- **Availability**: As the design phase of the new PCS Cargonaut solution started at Schiphol in 2019 and the development is still ongoing at the time this work is created, the solution and their data is considered up-to-date. Hence, all accountable people, for instance, the software architects and developers are available and accessible for interviews. In addition, necessary documents, like the ADD can be provided and can be consulted.
- **Confidentiality**: To create a working environment where data can be used with as few restrictions as possible, a confidentiality agreement is signed between the research institution (i.e. the university) and the interested organizations (i.e. CGI Nederland and Schiphol Group). However, if sensitive results need to be processed, the daily supervisor will be conducted. In general, the PCS solution is not considered sensitive and was therefore suitable to be selected.
- **Case size:** The case size of the PCS solution is considered very large. As already mentioned, the main part involves the integration (migration) of the Cargonaut datahub system into the Schiphol IT landscape. The solution relies on many existing ETO building blocks (7 ETO building blocks and 7 Schiphol solutions) and requires the development of specific solutions that are used for the PCS implementation (4 specific solutions). This volume makes it possible to obtain and study relevant data (e.g. various architectural principles). However, such a large volume also requires an extensive period of familiarization with the solution itself and its architecture by the researcher.
- **Development status:** There exist two environments for the PCS Cargonaut solution; (i) the Proof of Concept (PoC) environment and the (ii) Test environment. At the time this work is created there is no Live environment in production. Hence, the PCS case is considered as under development. This status means that the solution does not map the full range of functions, but only the core functionality. This core functionality focuses on receiving and processing only specific freight messages via specific communication protocols (i.e. the PoC environment has implemented three message types and uses MFT as the communication protocol).

- **Relevance**: The solution's ADD is structured into (i) business architecture with 5 principles, (ii) application architecture with 41 principles, (iii) technology architecture with 5 principles, and (iv) security architecture. The application architecture can be further divided into 4 general principles, 4 data principles, 17 integration principles, and 16 design principles. In total, the PCS Schiphol implementation is compliant with 51 architectural principles. This shows that the PCS solution affects all different layers of architecture. The Cargonaut integration is one key initiate of the IT & Data Strategy 2021-2023 of Schiphol. In addition, "Cargonaut enables the exchange of information across Schiphol's cargo community and is an essential part of day-to-day cargo operations at the airport". This highlights the importance of the Cargonaut solution for Schiphol's business strategy.
- **Completeness**: As mentioned by the Relevance criteria, the PCS solution complies with 51 architecture principles in all different architectural layers. This allows the consideration of a diverse range of principles and helps to identify a broad range of KPIs. Therefore, conclusions can be applied to other company-wide solutions and can be used beyond company borders.

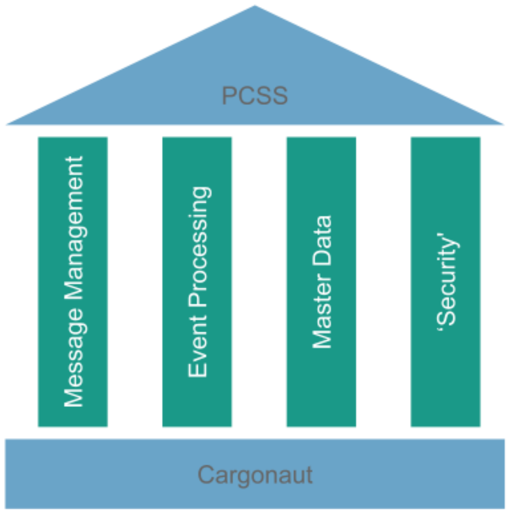
All criteria above positively contribute to the selection of the PCS Cargonaut solution as a research case. Only the large case volume and the PoC development status of the solution can be partly considered as negative aspects. However, both the case volume and the stage of development are considered to be a trade-off between (i) a wide range of available data and the extensive familiarization period, and (ii) the limited feature set and the availability of multiple architectural principles and the high contribution of the solution to the business strategy.

The next step involves the in depth analysis of the selected case. To do so, the Architectural Description Document (ADD) is used to extract the most relevant data for this research. This step extracts architectural principles on all different architectural levels (business, application, technology, security). Since this solution can be seen als large scale solution, an abstraction is necessary to divide the solution into high level components. This is done by using the ADD and structuring the solution into Conceptual Architectural Components. We use **triangulation** to validate our results and increase the precision of our extraction. We choose a **data source triangulation** by conducting **interviews** in addition to the data extraction of the documents. Hence, we used more than one data source and collected the same data at different occasions—written documents and the experts' knowledge.

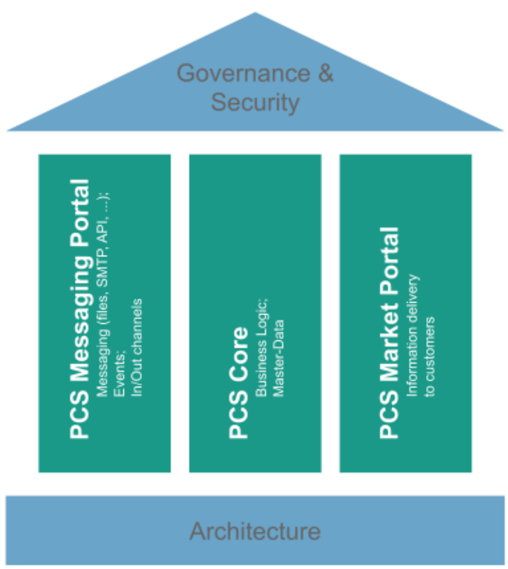
Research Execution – Phase #1

Intermediate Results

Conceptual Architectural Tiers INPUT



Conceptual Architectural Tiers OUTPUT



Research Execution – Phase #2

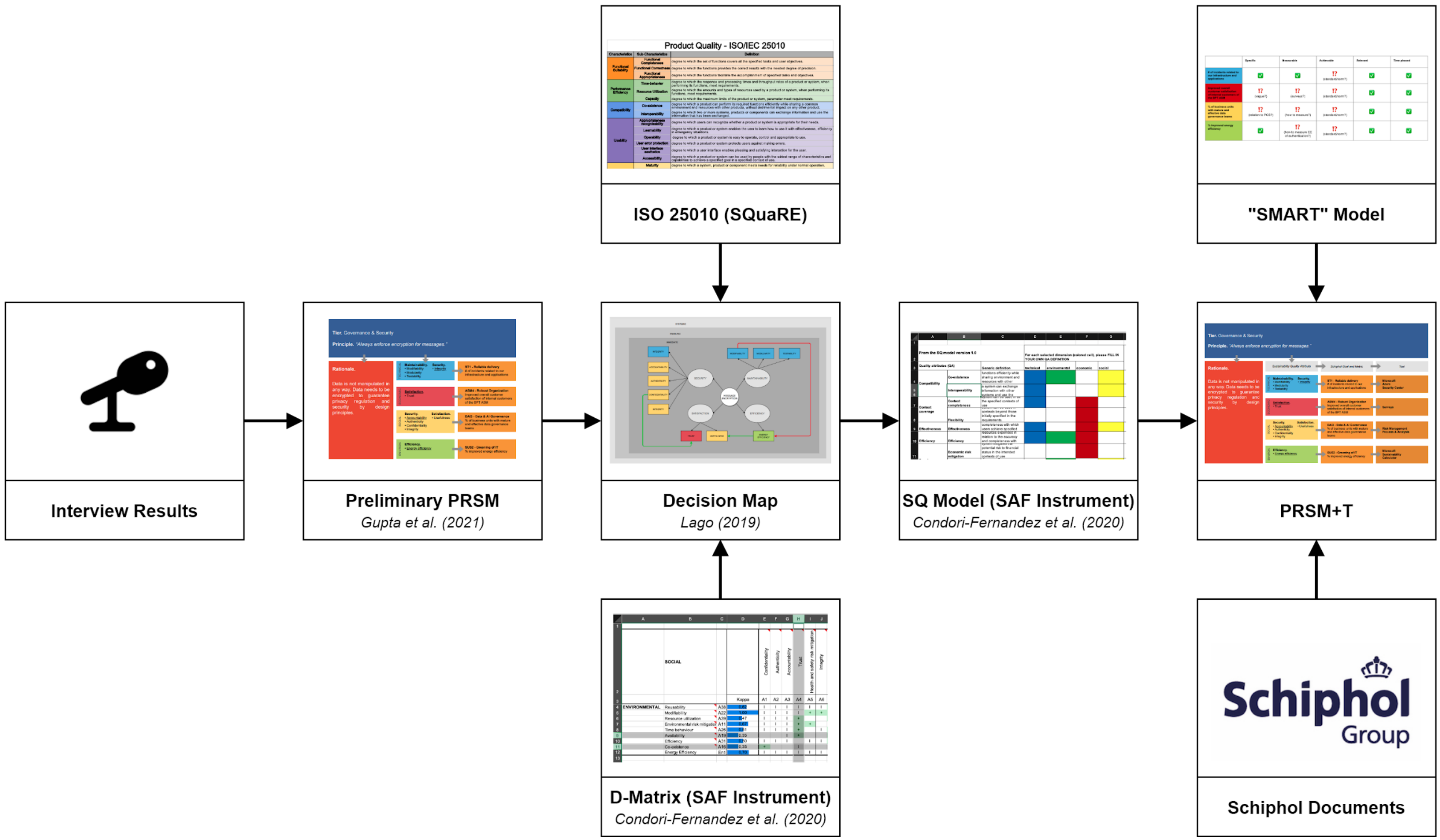
Intermediate Results

Conceptual Solution Tiers

The “Conceptual Architectural Components” were renamed into “Conceptual Solution Tiers” to better represent their actual meaning and eliminate confusions with “software components”. The tiers should represent the architectural structure, i.e. the internal division strategy of the PCS Cargonaut solution at Schiphol. For each tier, the driving software architectural principle was extracted by conducting the first round of interviews.

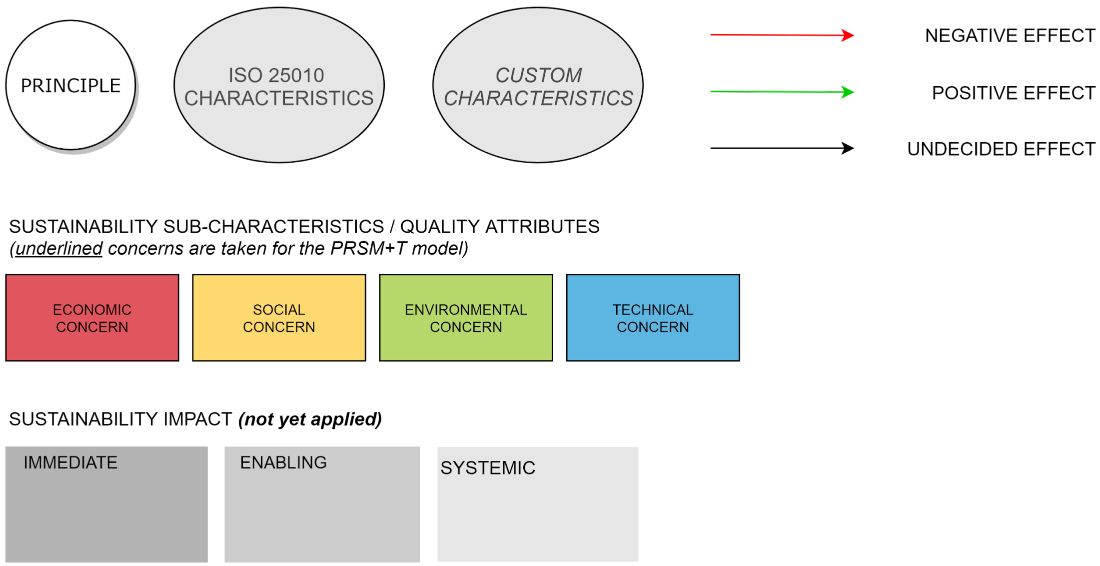
- **Governance & Security**
  Principle: *“Always authenticate flows and information requested by internal and external users.”*
- **PCS Core & Architecture**
  Principle: *“The system is made of loosely coupled components.”*
- **PCS Messaging Portal**
  Principle: *“SaaS goes above PaaS; PaaS goes above IaaS; IaaS goes above On-Premise.”*
- **PCS Market Portal**
  Principle: *“Use the Airport Service Bus (ASB) for sharing/exchanging of operational data between applications and parties where routing, filtering, data transformation(integration rules) or transport transformation capabilities are needed.”*

Process Pipeline



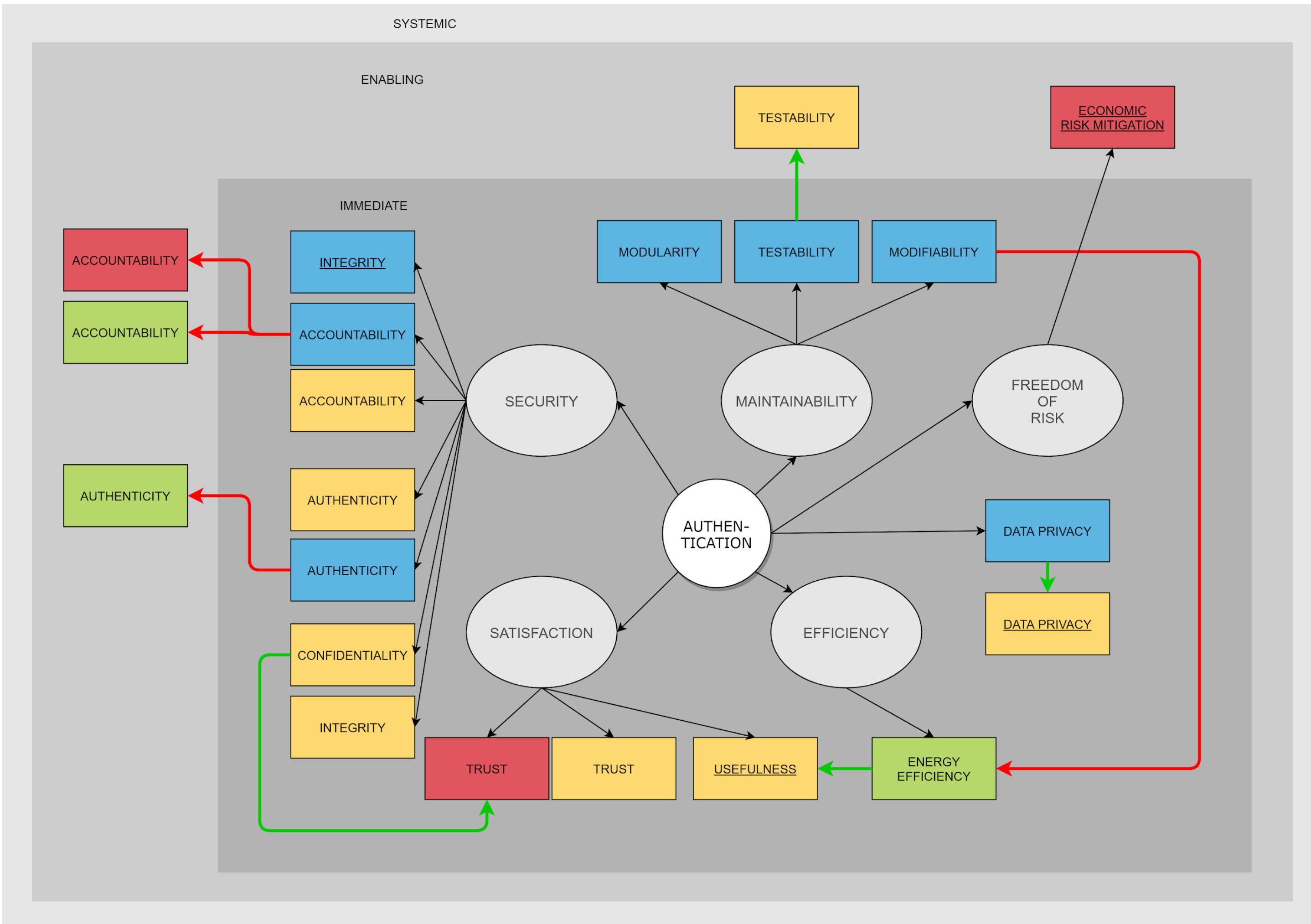
Decision Maps

Legenda



Governance & Security

Decision Map



SQ Model

QA		Generic definition	technical	environmental	economic	social
Security	Accountability	actions of an entity can be traced uniquely to the entity.	PCS system needs to log every action performed on network, hosting, storage and application level.	Due to logging activities, more storage needs to be allocated which leads to more energy consumption.	Due to logging activities, more storage needs to be allocated which causes more costs.	Flows and information can be traced uniquely to the internal or external users.
	Authenticity	the identity of a subject or resource can be proved to be the one claimed.	An Identity provider (IDP) like *Auth0*, or *Azure AD* needs to be introduced.	Additional SaaS solutions needs to be introduced which causes more energy consumption at the provider		Through authentication, the identity of the subject can be proved to be the one claimed.
	Confidentiality	system ensures that data are accessible only to those authorized to have access.				Only authorized PCS users can access and modify data if, and only if they have the permissions to access and modify the data
	Integrity	system prevents unauthorized access to, or modification of, computer programs or data.	PCS system always authenticate flows and information requested by internal and external stakeholders.			The PCS system prevents any unauthorized access and complies with the law regulations.
Satisfaction	Trust	stakeholders has confidence that a product or system will behave as intended.	n/a		Stakeholders have confidence that the PCS solution will behave as intended which will increase the economic revenues.	Stakeholders have confidence that the PCS solution behaves as intended which leads to trust in the system and the system will be used because it is trusted.
	Usefulness	user is satisfied with their perceived achievement of pragmatic goals.	n/a		n/a	The PCS stakeholders consider the PCS solution as useful and are satisfied with its implementation, even if they have to authenticate to get access.
Energy Efficiency	Energy Efficiency	N/A		Authentication requires computational extensive algorithms and hence more computational		

				power like CPU / memory		
Freedom from risk	Economic risk mitigation	system mitigates the potential risk to financial status in the intended contexts of use			By authenticating all flows and information, the risk of cyber attacks will be mitigated. Hence, the potential risk of financial loss is mitigated as well.	
Data Privacy	Data Privacy	privacy concerns arise wherever personally identifiable information is collected, stored, or used	Any communication and transmission, e.g. via REST, needs to be encrypted.			Personal data of all internal and external PCS users are transmitted encrypted and only accessible to those who have the permissions to the personal data.
Maintainability	Modifiability	system can be effectively and efficiently modified without introducing defects or degrading existing product quality.	Modifications on the PCS security and governance components do not introduce defects or security risks in the authentication component.	n/a	n/a	
	Modularity	system is composed of components such that a change to one component has minimal impact on other components.	Changes on other components have zero? Impact on the authentication component.			
	Testability	effectiveness and efficiency with which test criteria can be established for a system.	The implemented security and authentication mechanism can be tested efficiently and tested according to the Schiphol standards.			An effective and efficient test strategy leads to less effort in implementing test cases, hence, less effort for testing. Thus, the effectiveness and satisfaction of the tester increases.

PRSM+T

Tier. Governance & Security

Principle. *"Always authenticate flows and information requested by internal and external users."*

Rationale.

The PCS implementation consists of multiple different components which needs specific authentication and authorization capabilities.

By following the need-to-know principle, user access controls and authorisation procedures can be enforced. Its objective is to ensure that only authorised individuals gain access to information or systems necessary to undertake their duties.

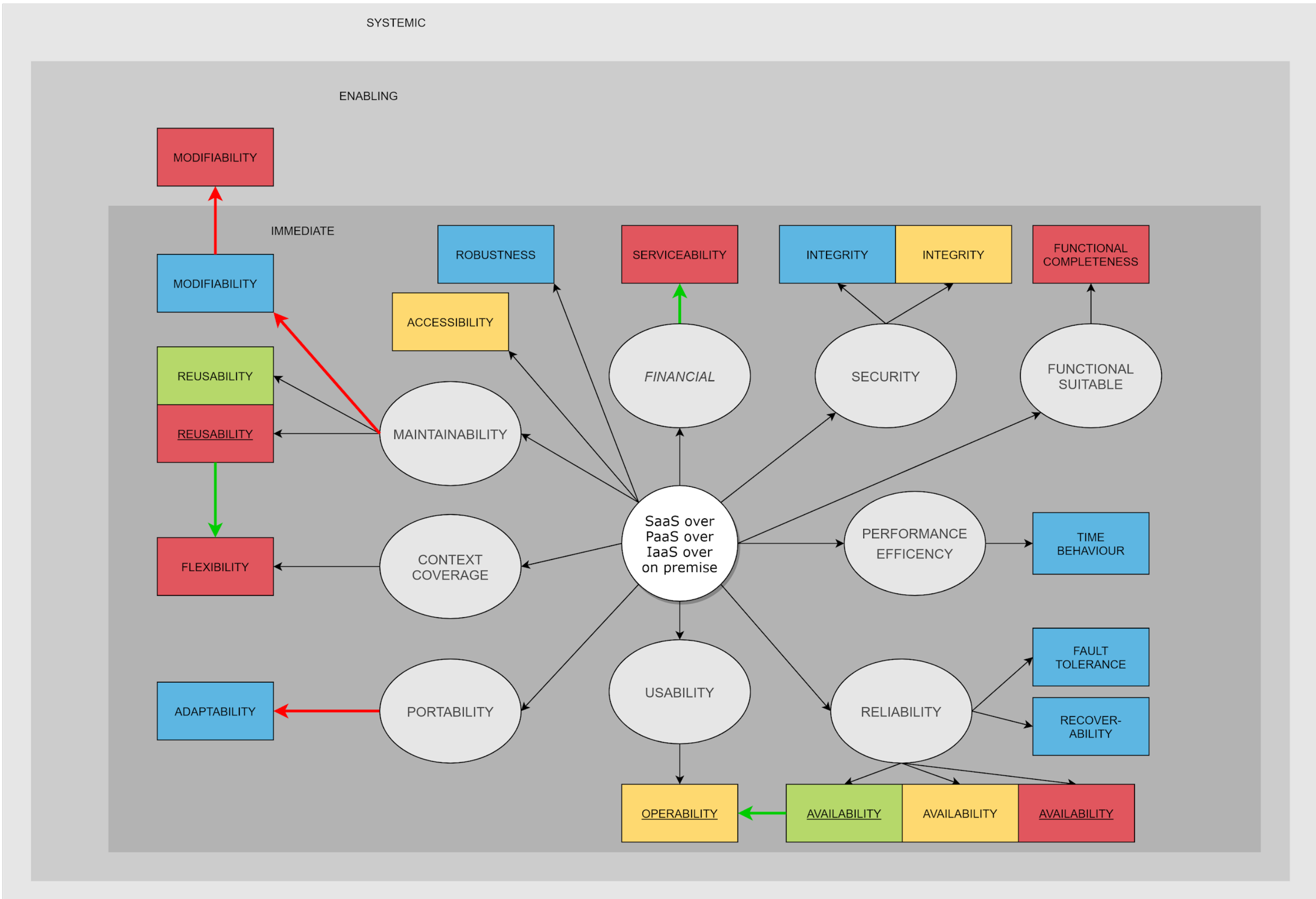
	Sustainability Quality Attribute	*Schiphol Goal and Metric*	*Tool*
TECHNICAL	**Maintainability.** • Modifiability • Modularity • Testability **Security.** • Integrity	**ST1 - Reliable delivery** # of security incidents within Schiphol's ICD	• **MS Azure Security Center** • **Qualys, Inc.** • **ICD@Schiphol (IBM Control Desk ???)**
ECONOMIC	**Satisfaction.** • Trust. **Freedom of Risk.** • Economic Risk Mitigation	• **DAI3 - Data & AI Governance** % of business units with mature & effective data governance teams • **CIO1 - Safe & Secure** Increase of the cyber maturity based on the ISF Framework.	• **Risk Management Process & Analysis** • **ISF Framework**
SOCIAL	**Security.** • Accountability • Authenticity • Confidentiality • Integrity **Data Privacy.** **Satisfaction.** • Usefulness	**ASM4 - Robust Organization** Improved overall customer satisfaction of internal and external customers (& maybe testers)	**Surveys among internal and external customers (& maybe testers)**
ENVIRON.	**Efficiency.** • Energy efficiency	**SUS2 - Greening of IT** % improved energy efficiency	**Microsoft Sustainability Calculator**

"SMART" Model

	Specific	Measurable	Achievable	Relevant	Time phased
Number of security incidents within Schiphol's ICD	✅	✅	⁉️ (baseline)	✅	✅
% of business units with mature and effective data governance teams	⁉️ (relation to PCS?)	⁉️ (how to measure?)	⁉️ (baseline)	✅	✅
Increase of the cyber maturity based on the ISF Framework. Long term goals is 3.5 for key value chains and 3 for foundational security	✅	✅	✅	✅	✅
Improved overall customer satisfaction of internal customers of the BPT ASM	⁉️ (vague?)	⁉️ (surveys?)	⁉️ (baseline)	✅	✅
% improved energy efficiency	✅	⁉️ (how to measure EE of authentication?)	⁉️ (baseline)	✅	✅

PCS Messaging Portal

Decision Map



SQ Model

QA		Generic definition	technical	environmental	economic	social
Maintainability	Modifiability	system can be effectively and efficiently modified without introducing defects or degrading existing product quality.	SaaS solutions can not be easily modified due to the provider dependencies. Modifications might be not possible due to provider restrictions.	n/a	The SaaS provider takes care of the modification, hence custom modifications might be either not possible at all or expensive.	
	Reusability	an asset can be used in more than one system, or in building other assets.		SaaS solutions can be reused by more than one system, even over customers and can be part of other solutions. Resources at the provider side can be shared.	If SaaS components can be reused across solutions, costs can be reduced.	
Context coverage	Flexibility	system can be used in contexts beyond those initially specified in the requirements.			SaaS solutions can be used in contexts beyond the PCS Cargonaut solution. SaaS solutions have the ability to match with business needs as they flow (Khanjani et al., 2014).	
Portability	Adaptability	system can effectively and efficiently be adapted for different or evolving hardware, software or usage environments.	The SaaS provider takes care of the hardware layer and its implementation. Hence, it is not possible to move the SaaS solution to an 'on-premise' solution/environment. However, the SaaS solution can be easily used in other environments on Schiphol's side and helps the organization to "act strategically not to react defensively" (Khanjani et al., 2014).			
Usability	Operability	system has attributes that make it easy to operate and control.				System changes can be made via the provider. No code changes have to be made to operate or control the system. This guarantees a fast and easy process for the customers eventually. ????? this is an assumption. a change can cause a breaking system there is no garantuee a change is backward compatible

Reliability	Availability	system is operational and accessible when required for use.	n/a	n/a	System, i.e. the SaaS solution needs to be highly available. If not, delays in the Cargo process can occur, leading to flight delays and thus enormous economical costs.	If the system is not available, the users and customers do not trust and do not use the solution.
	Recoverability	system can recover data affected and re-establish the desired state of the system in case of an interruption or a failure.	In case of an interruption or a failure on the provider side, the provider has to take care to recover the affected data and re-establish the desired state. SaaS solutions can also be used as a backup strategy to distribute viable data across multiple geographical locations and regions.		Data loss or an interruption would lead to enormous economical costs caused by flight delays and/or an interruption in the Cargo process pipeline.	
	Fault Tolerance	system operates as intended despite the presence of hardware or software faults.	Even in case of software or hardware faults on the provider side, the SaaS solution would/should operate as usual due to redundancy on the provider side. If, and only if the SaaS solution would have a hardware or software error, "the system would perform under acceptable and predictable bounds" (Khanjani et al., 2014).			
Performance Efficiency	Time Behaviour	response, processing times and throughput rates of a system, when performing its functions, meet requirements.	The time in which the SaaS solution processes the data and responds should meet the requirements.	n/a		
Functional suitability	Functional completeness	degree to which the set of functions covers all the specified tasks and user objectives.			The SaaS solutions need to cover the PCS needs as closely as possible. Otherwise, on-premise implementations are necessary which causes higher costs in addition to the SaaS costs.	
Security	Integrity	system prevents unauthorized access to, or modification of, computer programs or data.	The SaaS solution needs to guarantee that all data are stored, maintained, retrieved and transferred in an secured way, i.e. encryption. SaaS solutions are secured by the SaaS provider and prevent unauthorized access to, or modification of data. Nevertheless, it is possible to choose an own IDP provider.			SaaS solutions have to prevent unauthorized access to customer and sensitive data (e.g. for the customs).

Robustness	Robustness	Refers to the capability of the sytem to behave in an acceptable way in unexpected situations	Even if the SaaS solution has to deal with incomplete or inconsistent data, the service needs to be operable.			
Accessibility	Accessibility	system can be used by people with the widest range of characteristics and capabilities.				SaaS solutions are usable by users with different disabilities (Khanjani et al., 2014). This leads to access by many different user groups and with many different devices. In addition, the access to SaaS solutions are easier which decreases the barriers to the service.
Financial	*Serviceability*	(a.k.a. Supportability) System has attributes that make it easy to maintain beyond the software development cycle. It continues even when the software is no longer used.			The Life Cycle Management (LCM) for SaaS solutions is handled by the provider, hence, less support employees are necessary.	

PRSM+T

Tier. PCS Messaging Portal

Principle. *"SaaS goes above PaaS; PaaS goes above IaaS; IaaS goes above On-Premise."*

Rationale.

SaaS solutions help to reduce the cost and maintenance overhead of running cloud services.

The technical knowledge does not need to be at company level and can be pass to the provider. This minimizes the risk of incidents.

It is necessary to ensure that the cloud solution comply with the company infrastructure and can be integrated.

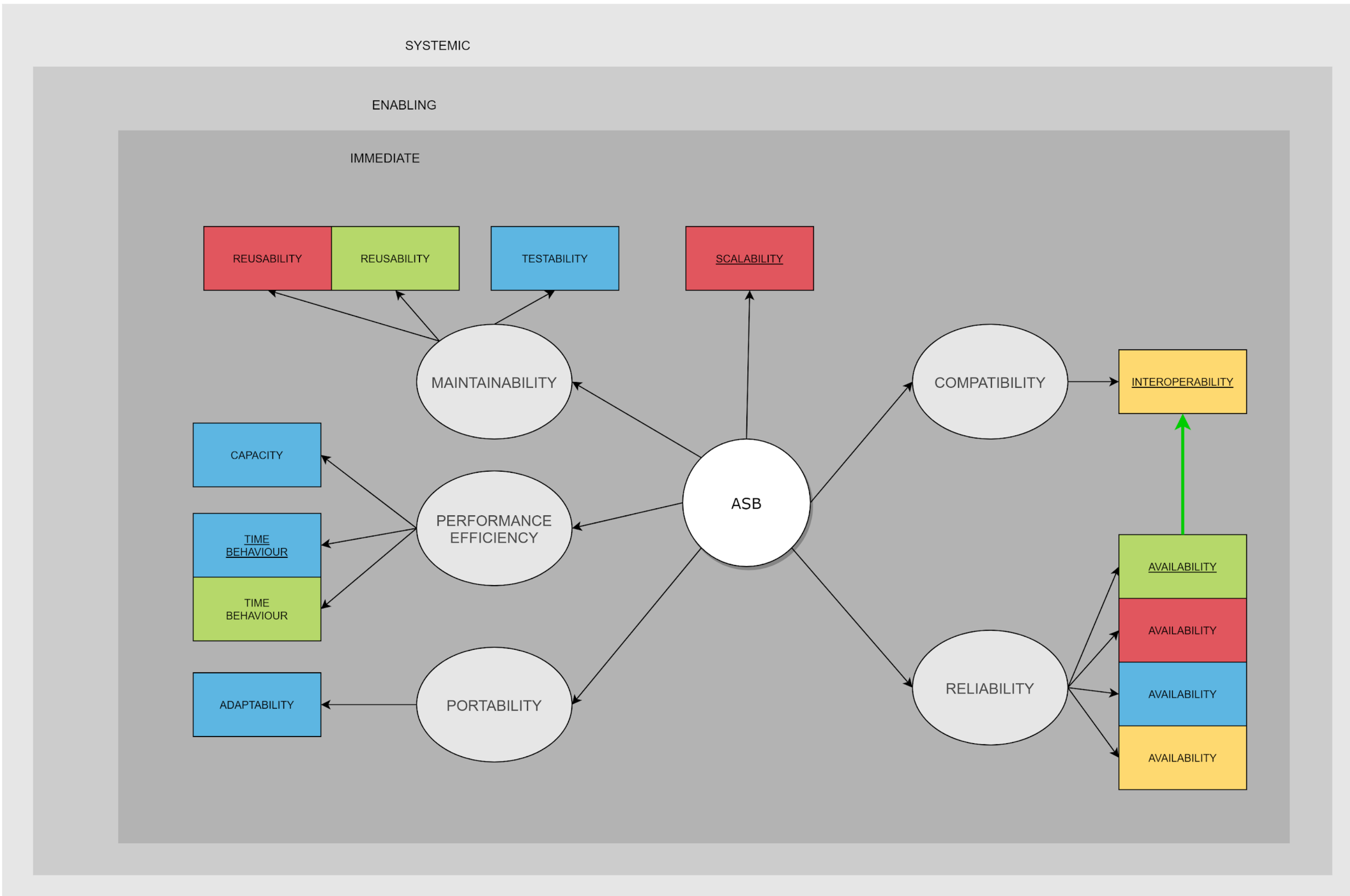
	Sustainability Quality Attribute	*Schiphol Goal and Metric*	*Tool*
TECHNICAL	**Portability.** • Adaptability **Security.** • Integrity **Maintainability.** • Modifiability	**ETO2 - Reliable Delivery** *[CUSTOM]* Number of changes	**SonarQube**
ECONOMIC	**Maintainability.** • Reusability • Modifiability **Context Cov.** • Flexibility **Financial.** • Serviceability **Reliability.** • Availability	**ETO1 - Realize enabling technology outlook** # of ET building blocks that has been successfully implemented	**SPLUNK ???**
SOCIAL	**Usability.** • Operability **Satisfaction.** • Usefulness **Security.** • Integrity **Reliability.** • Availability	**OPS3 - Business change** Number of API calls	**Schiphol API Dashboard**
ENVIRON.	**Maintainability.** • Reusability **Reliability.** • Availability	**ETO2 - Reliable Delivery** Up-time for key platforms	**SPLUNK ???**

“SMART” Model

	Specific	Measurable	Achievable	Relevant	Time phased
[Custom] Number of changes	✅	⁉️ (how?)	⁉️ (baseline)	⁉️ (is it relevant?)	✅
Number of ET building blocks that has been successfully implemented	⁉️ (rather vague)	⁉️ (how?)	⁉️ (baseline)	✅	✅
Number of API calls	✅	✅	⁉️ (baseline)	✅	✅
Up-time for key platforms	✅	✅	⁉️ (baseline)	✅	✅

PCS Market Portal

Decision Map



SQ Model

QA		Generic definition	technical	environmental	economic	social
Maintainability	Reusability	an asset can be used in more than one system, or in building other assets.		Reusing the ASB as a building block beyond PCS boundaries saves resources.	Reusing the ASB as a building block beyond PCS boundaries reduces implementation and modification costs.	
	Testability	effectiveness and efficiency with which test criteria can be established for a system.	The PCS integration / connection to the ASB should be effectively and efficiently testable without affecting other systems.			
Performance efficiency	Capacity	the maximum limits of a product or system parameter meet requirements.	The PCS integration / connection to the ASB does not affect the maximum limits of other systems parameters and meets the PCS parameter requirements			
	Time behaviour	response, processing times and throughput rates of a system, when performing its functions, meet requirements.	The PCS integration / connection to the ASB does not affect the response, processing times and throughput rates of other systems and meets the PCS requirements.	The PCS integration / connection to the ASB meets the resource requirements regarding the response, processing times and throughput rates. Hence, the ASB meets the environmental requirements.		
Portability	Adaptability	system can effectively and efficiently be adapted for different or evolving hardware, software or usage environments.	The ASB (2.0) can be effectively and efficiently be adapted for different systems and environments beyond the PCS boundaries and can be hosted on different cloud providers by using the ASB as a cloud solution.			
Reliability	Availability	system is operational and accessible when required for use.	The ASB is operational and accessible when the PCS solution needs it. Availability is ensured by the cloud provider (ASB 2.0) and	n/a	The ASB is operational and accessible when the PCS solution needs it. If so, the customers are able to use the PCS solution	The ASB is operational and accessible when the PCS solution needs it. If so, internal and

			guaranteed by using redundancy.		and hence, economic aspects are satisfied. However, downtime would cause irregulations in the Cargo process, would lead to flight delays and cause enormous economic costs.	external users, especially the customs can use it which leads to a controlled Cargo process.
Scalability	Scalability	the ability of a computing process to be used or produced in a range of capabilities	Scalability is achieved by using OS-level virtualization methods, i.e. containerization. This technology allows dynamic scaling of resources.	Containerization allows dynamic scaling of resources which leads to better energy efficiency.	The ASB (2.0) has the ability to scale and can be used across PCS boundaries. By using the ASB for a variety of systems without implementing a new scalability mechanism, costs are mitigated.	
Compatibility	Interoperability	a system can exchange information with other systems and use the information	n/a			By using the ASB for the PCS solution, information can be received and exchanged with other systems and other users beyond the internal and external stakeholders.

PRSM+T

Tier. PCS Market Portal

Principle. "*Use the Airport Service Bus (ASB) for sharing / exchanging of operational data between applications and parties where routing, filtering, data transformation (integration rules) or transport transformation capabilities are needed.*"

Rationale.

The ASB is an integration platform, because it adds functionalities to integrate two or more known systems.

ASB adds functionality like routing, transformation, aggregation, throttling, basic reliable messaging and user management.

Beyond, the ASB causes more integration work. This need is due to the increased data exchange when the # of connected applications increases.

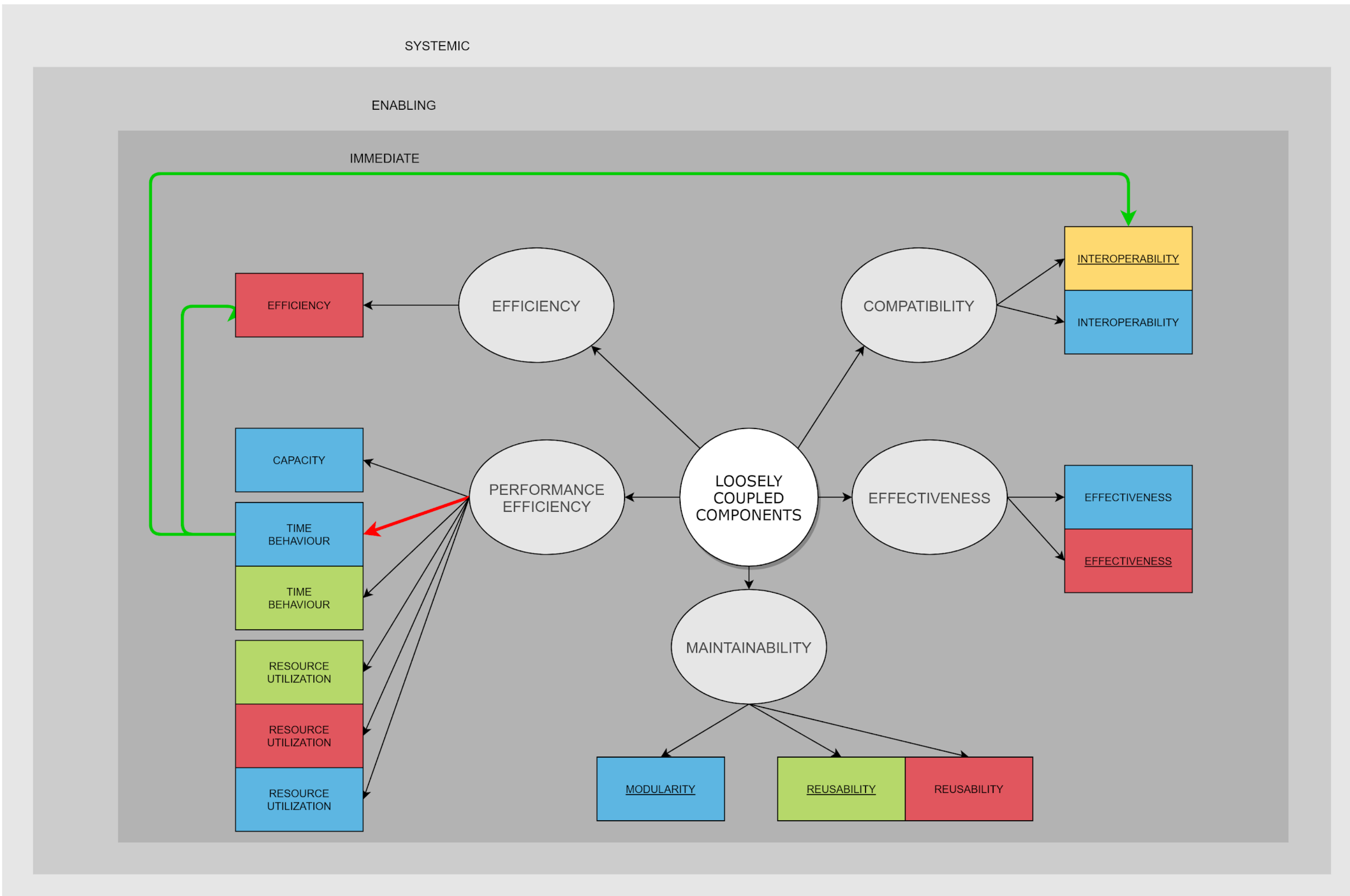
	Sustainability Quality Attribute	*Schiphol Goal and Metric*	*Tool*
TECHNICAL	**Reliability.** • Availability **Maintainability.** • Testability **Perfor.-Effic.** • Capacity • Time behaviour **Portability.** • Adaptability	**ETO2 - Reliable Delivery** *[Custom (ISO)]* Throughput	**SPLUNK ???**
ECONOMIC	**Reliability.** • Availability **Scalability**	**Solid Foundation - Connectivity** *[Custom]* system performance grows with adding more resources	**???**
SOCIAL	**Reliability.** • Availability **Compatibility.** • Interoperability	**ST3 - One Foundation** *[Custom (ISO)]* Number of Data Formats	**SPLUNK ???**
ENVIRON.	**Reliability.** • Availability **Perfor.-Effic.** • Time behaviour	**ETO2 - Reliable Delivery** Up-time for key platforms	**SPLUNK ???**

"SMART" Model

	Specific	Measurable	Achievable	Relevant	Time phased
[Custom (ISO)] Throughput	✅	✅	⁉️ (baseline)	✅	⁉️ (no?)
[Custom] system performance grows with adding more resources	⁉️ (rather vague)	⁉️ (how?)	⁉️ (baseline)	✅	⁉️ (no?)
[Custom (ISO)] Number of Data Formats	⁉️ (rather vague)	⁉️ (how?)	⁉️ (baseline)	⁉️ (is it relevant?)	⁉️ (no?)
Up-time for key platforms	✅	✅	⁉️ (baseline)	✅	✅

PCS Core & Architecture

Decision Map



SQ Model

QA		Generic definition	technical	environmental	economic	social
Maintainability	Modularity	system is composed of components such that a change to one component has minimal impact on other components.	PCS solution is composed of components such that a change to one component (e.g., Event Processing) has minimal/no impact on other components. This is achieved by using OS-level virtualization, i.e. containerization.	To maintain communication between the components, more communication effort is necessary, which leads to more resource utilization.	Changes on small and loosely coupled components can be done efficiently and easy. Developers have less effort by implementing such changes, thus, economic costs are reduced.	==> Usefulness
	Reusability	an asset can be used in more than one system, or in building other assets.		PCS components can be used in more than one system. E.g. the Event Processing building block can be used beyond PCS boundaries. This saves environmental resources.	Reusable PCS components beyond system boundaries would save economical costs.	
Satisfaction	Usefulness	user is satisfied with their perceived achievement of pragmatic goals.		n/a	n/a	Changes can be faster delivered which leads to more satisfaction for developers as well as customers.
Performance efficiency	Capacity	the maximum limits of a product or system parameter meet requirements.	One component meets the requirements regarding the maximum limits of a component.			
	Resource utilization	the amounts and types of resources used by a system, when performing its functions, meet requirements.	Loosely coupled components are achieved by using OS-level virtualization, i.e. containerization.	A PCS component, i.e. the container, allocates resources only for its own needs and does not affect the resources of other components. This saves environmental resources.	By using technologies like containerization, resources can be shared across different components/containers which leads to less expenses.	
	Time behaviour	response, processing times and throughput rates of a system, when performing its functions, meet	The, e.g., Queue time of one component has to be constant and does not affect other components. Instead of using system			

		requirements.	(class/method) calls, API calls are necessary which leads to higher latency.			
Effectiveness	Effectiveness	accuracy and completeness with which users achieve specified goals.	Each PCS component can perform its tasks precisely and completely on its own, without using other components by making use of Containerization.		Each PCS component can perform its tasks precisely and completely on its own, without using other components.	n/a
Efficiency	Efficiency	resources expended in relation to the accuracy and completeness with which users achieve goals	n/a	n/a	The allocated resources for a component expand in proportion to the accuracy and completeness with which the PCS requires them to meet its objectives without affecting other components.	
Compatibility	Interoperability	a system can exchange information with other systems and use the information	PCS components can exchange information with other systems and beyond system boundaries.			PCS components can receive and exchange information with other systems and other users beyond the internal and external stakeholders.

PRSM+T

Tier. PCS Core and Architecture

Principle. "*The system is made of loosely coupled components.*"

Rationale.

There are many different types of communication protocols currently supported to deliver Cargo related messages or request Cargo related information by customers. To be able to handle all kinds communication, loosely coupled components are necessary.

E.g., the PCS Core System is implemented in Sub-Components and is loosely coupled to the Business Engine which is implemented outside of the Core. In addition, the responsibilties of the components are distributed across different layers and messages (i.e., API) are used for communication between the layers.

	Sustainability Quality Attribute	*Schiphol Goal and Metric*	*Tool*
TECHNICAL	**Modularity.** PCS solution is composed of components such that a change to one component (e.g., Event Processing) has minimal/no impact on other components. This is achieved by using OS-level virtualization, i.e. containerization.	**ETO1 - Realize enabling technology outlook** # of ET building blocks that has been successfully implemented; # of components build; # of containers	**Application Performance Monitoring (APM)**
ECONOMIC	**Effectiveness.** Each PCS component can perform its tasks precisely and completely on its own, without using other components.	**Standardized and scalable technology landscape** Costs per changes;	**JIRA**
SOCIAL	**Usefulness.** Changes can be faster delivered which leads to more satisfaction for developers as well as customers.	**ASM4 - Robust Organization** • Improved overall satisfaction of internal and external customers • [*Custom*] component delivery time	**• Surveys among internal and external customers • JIRA**
ENVIRON.	**Reusability.** PCS components can be used in more than one system. E.g. the Event Processing building block can be used beyond PCS boundaries. This saves environmental resources.	**ETO1 - Realize enabling technology outlook** # of ET building blocks that has been successfully implemented; # of components build; # of containers	**Application Performance Monitoring (APM)**

“SMART” Model

	Specific	Measurable	Achievable	Relevant	Time phased
Number of ET building blocks that have been successfully implemented	⁉️ (rather vague)	⁉️ (how?)	⁉️ (baseline)	✅	✅
Number of components build	✅	✅	⁉️ (baseline)	✅	✅
Number of containers	✅	✅	⁉️ (baseline)	✅	✅
Costs per change	✅	✅	⁉️ (baseline)	✅	✅
Improved overall satisfaction of internal and external customers	⁉️ (vague?)	⁉️ (surveys?)	⁉️ (baseline)	✅	✅
[Custom] Component delivery time	✅	⁉️ (how?)	⁉️ (baseline)	✅	✅
Number of ET building blocks that has been successfully implemented	⁉️ (rather vague)	⁉️ (how?)	⁉️ (baseline)	✅	✅

Research Execution – Phase #3

Interviews

⇒ see interviews in replication package

Results

⇒ see DrawIO sheet for final

- PRSM+T models
- Decision Maps

Focus Group

⇒ see Appendix D

Spider Charts

⇒ see replication package

- Python scripts for all architectural principles
- Python scripts for mock-up normalized version

Appendix A - Case Study Checklist

This checklist is derived from Runeson and Höst (2008). We only use the “Researcher’s Checklist” since we are interested only in conducting a case study instead of reviewing a case study.

No.	Description	Check	Notes
Case study design			
1	What is the case and its units of analysis?	✅	
2	Are clear objectives, preliminary research questions, hypotheses (if any) defined in advance?	✅	No hypotheses defined
3	Is the theoretical basis—relation to existing literature or other cases—defined?	✅	
4	Are the authors’ intentions with the research made clear?	✅	
5	Is the case adequately defined (size, domain, process, subjects...)?	⁉️	Size: could be too huge
6	Is a cause–effect relation under study? If yes, is it possible to distinguish the cause from other factors using the proposed design?	**n/a**	
7	Does the design involve data from multiple sources (data triangulation), using multiple methods (method triangulation)?	✅	
8	Is there a rationale behind the selection of subjects, roles, artifacts, viewpoints, etc.?	✅	
9	Is the specified case relevant to validly address the research questions (construct validity)?	✅	Moved to report
10	Is the integrity of individuals/organizations taken into account?	✅	
Preparation for data collection			

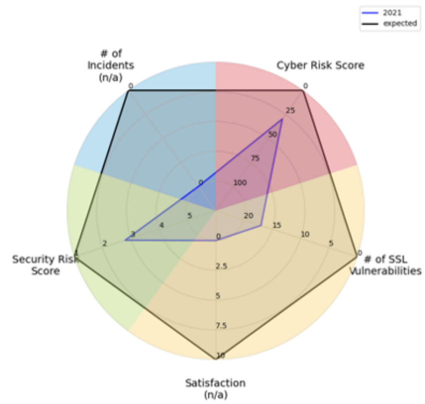
11	Is a case study protocol for data collection and analysis derived (what, why, how, when)? Are procedures for its update defined?	✅	
12	Are multiple data sources and collection methods planned (triangulation)?	✅	
13	Are measurement instruments and procedures well defined (measurement definitions, interview questions)?	✅	All interview structures reviewed by the daily supervisor; transcripts are created afterwards
14	Are the planned methods and measurements sufficient to fulfill the objective of the study?	✅	Review by both daily supervisor and first supervisor.
15	Is the study design approved by a review board, and has informed consent obtained from individuals and organizations?	✅	Review by both daily supervisor and first supervisor.
Collecting Evidence			
16	Is data collected according to the case study protocol?	✅	See study design
17	Is the observed phenomenon correctly implemented (e.g. to what extent is a design method under study actually used)?	**n/a**	
18	Is data recorded to enable further analysis?	⁉️	Partly. Transcripts are created from the interviews and focus group. However, the raw data had to be omitted due to safety and security reasons.
19	Are sensitive results identified (for individuals, the organization or the project)?	⁉️	Yes. see above.
20	Are the data collection procedures well traceable?	✅	
21	Does the collected data provide ability to address the research question?	✅	
Analysis of collected data			
22	Is the analysis methodology defined, including roles and review procedures?	✅	
23	Is a chain of evidence shown with traceable inferences from data to research questions and existing theory?	✅	Intermediate results provided

24	Are alternative perspectives and explanations used in the analysis?	✅	???
25	Is a cause–effect relation under study? If yes, is it possible to distinguish the cause from other factors in the analysis?	**n/a**	
26	Are there clear conclusions from the analysis, including recommendations for practice/further research?	✅	
27	Are threats to the validity analyzed in a systematic way and countermeasures taken? (Construct, internal, external, reliability)	✅	
Reporting			
28	Are the case and its units of analysis adequately presented?	✅	
29	Are the objective, the research questions and corresponding answers reported?	✅	
30	Are related theory and hypotheses clearly reported?	**n/a**	
31	Are the data collection procedures presented, with relevant motivation?	✅	
32	Is sufficient raw data presented (e.g. real life examples, quotations)?	✅	Interview quotations, yes; data for spider charts
33	Are the analysis procedures clearly reported?	✅	
34	Are threats to validity analyses reported along with countermeasures taken to reduce threats?	✅	
35	Are ethical issues reported openly (personal intentions, integrity issues, confidentiality)?	⁉️	Partly; only security and safety related issues
36	Does the report contain conclusions, implications for practice and future research?	✅	
37	Does the report give a realistic and credible impression?	✅	
38	Is the report suitable for its audience, easy to read and well structured?	✅	

Appendix D - Focus Group

For each set of PRSM+T model and Spider Chart (e.g.)

#1)

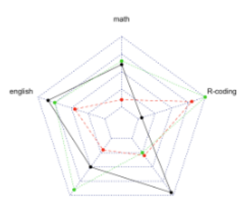


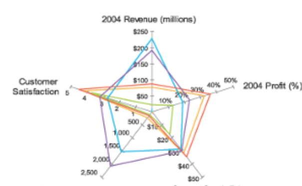
Software Sustainability

Schiphol
Welcome to Amsterdam Airport

1. Is the final model well defined and can you confirm the combination with the Spider Plot?
2. What does the data (Spider Plot) mean to you?
3. What are potential problems in using or understanding the model and the plot?

Spider Charts

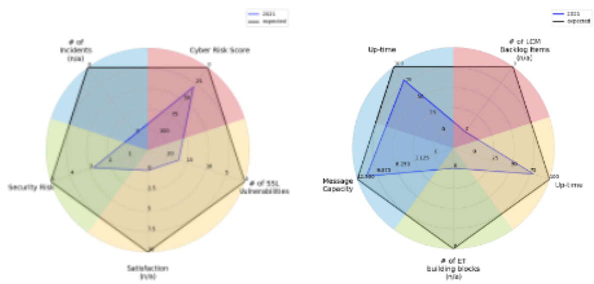




- **What is your opinion about Spider Charts?**
- **Do you use Spider Charts @Schiphol and for what?**
- **What is your opinion about Spider Charts with different scales (right plot)?**

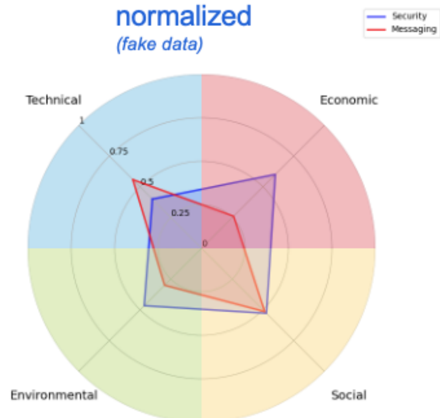
Schiphol
Welcome to Amsterdam Airport

different scales



vs.

normalized
(fake data)



- **How do you interpret the normalized Spider Plot?**
- **Does the normalized Spider Plot have advantages?**
- **How would you define the actual *impact* ?**

Software Sustainability

Schiphol
Welcome to Amsterdam Airport

Do you have final remarks or questions?

Afbeelding toevoegen

22

References

Brereton, Pearl, et al. “Using a Protocol Template for Case Study Planning.” 2008, doi:10.14236/ewic/ease2008.5.

Gupta, Sarthak, et al. “A Framework of Software Architecture Principles for Sustainability-Driven Design and Measurement.” *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, 2021, doi:10.1109/icsa-c52384.2021.00012.

Lago, Patricia. “Architecture Design Decision Maps for Software Sustainability.” *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, 2019, doi:10.1109/icse-seis.2019.00015.

Runeson, Per, and Martin Höst. “Guidelines for Conducting and Reporting Case Study Research in Software Engineering.” *Empirical Software Engineering*, vol. 14, no. 2, 2008, pp. 131–164., doi:10.1007/s10664-008-9102-8.