CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

A Framework for Guidance of API Governance: A Design Science Approach

Bachelor of Science Thesis in Software Engineering and Management

Emanuel Lourenço Marcos
Raphael Puccinelli de Oliveira

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

This paper presents a framework for guidance over API governance.
It also proposes a framework implementation process, based on decision models.
In order to evaluate and motivate the framework components,
the data collection process was conducted in cooperation with five distinct big scale organizations.
The resulting framework is our contribution to the research field of API governance.

Emanuel Lourenço Marcos
Raphael Puccinelli de Oliveira

Supervisor: Jennifer Horkoff
Examiner: Agneta Nilsson

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

A Framework for Guidance of API Governance: A Design Science Approach

Emanuel Lourenço Marcos
Software Engineering and Management Program
Department of Computer Science and Engineering
Gothenburg University, Gothenburg, Sweden
E-mail: guslouem@student.gu.se

Raphael Puccinelli de Oliveira
Software Engineering and Management Program
Department of Computer Science and Engineering
Gothenburg University, Gothenburg, Sweden
E-mail: guspucra@student.gu.se

***Abstract*—Application Programming Interface (APIs) provide access to business assets such as data and services. Decisions associated with the implementation of new APIs or change to existing ones may lead to substantial implications on the business value that APIs provide. As such, governance over APIs should be one of the main concerns for organizations that employ APIs as part of their business. We conducted a design science research aiming to identify the relevant aspects and strategies of API governance. As a result, we present a conceptual framework for guidance over API governance, in a research conducted as part of an on-going project and in collaboration with several large-scale companies that make extensive use of APIs. Through the conceptual framework we present the identified relevant aspects, strategies and a proposal for the establishment of a governance board.**

***Keywords*-API; aspects; governance; guidance; strategies.**

I. Introduction

A. Application Programming Interface

Application Programming Interface (**APIs**) are commonly used by software engineers to combine their own software with existing libraries [1]. APIs allow the interfacing of different pieces of software together all the while reducing the dependencies linking them [2]. Most software operates in a software ecosystem. In this context, APIs enable the interface between an asset and the users of the asset [3]. This raises the importance of standardized access and stability [4]. Additionally, APIs can define a clear boundary of the domain, emphasizing control and visibility over traffic on the platform where they are deployed, decoupling implementation details from access to the technologies [5]. API design and evolution must evolve along with the system to provide the expected service [3]. The decision process behind API design, creation, development or modifications should be done while fully accounting for the needs the API must fulfill.

B. Motivation

Governance often relates to management, and governance of APIs is of paramount importance. Without proper governance, issues might lead to interrupted access or erroneous functionality, negatively impacting the business [6]. APIs assume a high relevance in digital business. Yet, there is a perceivable gap in the efforts put forward to understand and improve API management. Poor API governance can eventually lead to undesired consequences, such as a decrease in the product's security or quality. In a business context, there is an increasing focus on using APIs as access points to encapsulated services. As such, in order to provide the value their customers expect, and to avoid the above-mentioned consequences, these APIs must be carefully developed and managed [7].

A critical analysis of API governance in different contexts enables comparisons to be made. Positive and negative aspects can be identified, what works and what does not work so well. The criticality of API governance can be summarized as follows [4]: First, empower the right people to do governance over APIs in an organizational context. Second, evaluate and analyze the effects that change might bring to existing API governance, ensuring that functionality and access to an asset are consistent and maintained. Additionally, these changes should be tracked and analyzed to maintain an audit over the governance of the API.

C. Context

This thesis is part of a research project in the environment of ecosystem-driven development, focusing on API strategies. The research project is conducted at the University of Gothenburg and Chalmers Software Center [8]. The aim is to enable industrial partners to build an API strategy involving internal actors and external stakeholders, by taking into consideration the needs of partners that deal with different dimensions of API strategy [8]. As part of a continuous project, five companies invested in API management and strategies cooperated in this research which allowed for the understanding of each companies' context, situation, and needs in terms of API governance. From this cooperation and along with related literature, we will investigate what are the most relevant aspects of API governance, and what strategies to employ in API governance. Through an empirical approach and related literature analysis, we will propose a framework for API governance, consisting of relevant governance aspects for evaluation, governance strategies, and their trade-offs.

D. Research Questions

The main research questions we aim to answer are:

- **RQ.1:** What are the relevant aspects of API governance?

- **RQ.2:** What are relevant strategies of API governance?
- **RQ.3:** How can API governance guidelines be captured and presented?

To answer the first research and second research questions, we will draw from relevant literature to identify the relevant aspects and strategies. Interaction with the collaborating companies allows for the understanding of how well these translate to real-life scenarios. Moreover, it allows us to identify possible missing aspects and strategies. Conclusions can then be drawn, such as trade-offs and relevancy, by contrasting the literature and real-life scenarios. To answer the third research question, a framework for guidance over API governance is proposed, encapsulating the collected information. The elected design science methodology approach for this thesis allows for the researchers to iterate on the solution to the output of RQ.3.

E. Structure

The remainder of this paper is structured in the following manner. Section II provides insight on existing literature on the problem domain. Section III details the research methodology employed to conduct data collection, analysis and how to present it. Methodology for the evaluation of the resulting artifact is also presented. Section IV introduces the final artifact. Section V discuss the collected data and details the reasoning for the final outcome. Section VI discusses identified threats to the validity of this research. Section VII contains our final thoughts.

II. Related Literature

Research literature is scarce in the field of API governance. In particular, there is a lack of research that focuses on understanding, evaluation, and development of frameworks for API governance [4]. This thesis is inserted in the research project conducted at Software Center. Previous work within the project aims to provide an analytic framework for designing and managing API systems. This research aims to produce a solution for guidance over API governance. As such, we start by analyzing the existing data from previous work in the project. The produced solution is based upon further investigation on how governance is done at each company, as well as by drawing from relevant literature.

A. Software Center Background Work

Business-centric efforts have prompted an increasing focus on the topic of value in system and software engineering [9]. In order to capture the concept, several modeling approaches have been put forward. Horkoff et al. [9] e^3 apply value modeling to a cross-company comparative case study, where internal and external value of strategic APIs is modeled. The authors propose e^3 models for each of the five collaborating companies, communicating positive and negative findings.

Horkoff et al. [10] defend that APIs are more valuable than simple technical functions. Recently, the perception of APIs has shifted from an interpretation of APIs as the means to separate implementation from function calls, to a view that promotes that APIs can be an integral part of a business plan for software-intensive companies [10]. As part of a general framework for strategic API analysis, existing modeling techniques are utilized to evaluate software API and planning from a strategic business point of view. Not only are commonly agreed upon benefits from modeling kept, but the authors also discovered API-specific benefits, such as better understanding of API work-flows through work-flow modeling, as well as API planning and development motivation which can be captured by ecosystems models.

Lindman et al. [11] discuss emerging perspectives to API strategy, resulting in a framework that summarizes and unifies API strategies. Several layers for API strategies are discussed, such as: *Domain layer*, where needs and events supported by the API are found; *App usage layer* where user-visible features reside; *API layer* where access to the business asset lies; and *Business asset layer* relating to concerns over business assets pertaining to the company, such as product properties, algorithms, and data. Boundary objects between the layers can exert influence in an organization's API strategy. Organizations look at artifacts when making decisions, as API strategies are based on identified boundary objects [11]. API strategies are interested in the commonly found boundary artifacts. First, *Use cases* which outline how actors achieve goals based on existing features. API strategies are concerned with which use cases exists, how do these change, and what impact does that change bring. *API specification*, an agreement between App developer and API provider. API strategies questions the impact of or refusal to change. Finally, *API model* contains important aspects of available business assets. API strategy is concerned on whether all important aspects are included, the effectiveness of the model in allowing for App software that covers important use cases to be developed, and model efficiency which allows relevant use cases to be addressed while considering optimal resource usage.

API governance is considered an important issue in API strategy, due to aspects such as access control, openness, and resource management. Different layers bring different governance issues, leading to companies to decide which governance strategies can better support their business goals at each layer [11]. Examples of such strategies are access control to business assets or selected parts of assets, the inclusion of invested partners in API planning and evolution, attracting outside contributors by releasing parts of the API as open source, or bandwidth consumption policies to API clients. Finally, a decoupling between layers and strategy means that companies do not have to employ the same strategy for all layers [11].

B. API Governance in Software Development

Governance in the information technology (**IT**) domain can bring business value to an organization [12]. Bannerman [13] proposes a meta-perspective of governance in software governance. Software development as a subset of the IT domain is continuously pushed forward in its advancements and search

for improvement opportunities by the increasing demand for faster delivery of better, broader and more advanced software. Much like in the governance over APIs topic, little research exists on the topic of software development governance, with existing research rather focusing on higher organizational stack [13].

Governance can be seen as a multi-dimensional concept, however, it should not be mistaken as management despite sharing common aspects, such as transparency and accountability [13]. Governance embraces diverse elements and tends to be interpreted from two distinct points of view. The functionally perspective relates to what governance does, focusing on governance as a verb; structural perspective refers to what the governance looks like, how it is modeled, and where governance is viewed as a noun. This thesis proposes a framework for how to conduct API governance, as such a significant focus is put on the structural perspective. With that said, it is important to stress that a focus on either perspective at the cost of the other can lead to a biased and narrow interpretation of governance [13]. Governance can also be interpreted as strategy, safeguarding the established goals and enabling vision, direction, and capabilities of the organization [12]. On the other hand, governance is often perceived as a process, where people are given decision rights and those rights are kept in check, all the while reviewing and a regulating the assignments of such rights and processes [12]. Bannerman [13] discusses the distinction between active and passive governance. Active promotes directional guide in an interventionist manner, while passive takes on a more distant approach. Absolute versus situational governance is also put forward, relating to the context and characteristics of the environment and organization in which governance is applied. In this thesis, the proposed framework will be drawn not only from literature but also from the interaction with five distinct companies. While the framework here proposed takes on a generic governance approach, it is more likely to bring greater value to an organization when further tailored to the organization's business model, context, and size.

C. API Governance

API governance relates to API implementation, management and deployment control in a business context [10]. Research on APIs focuses mostly on API design, language or service specific APIs, and development best practices [14], [15]. However, in industrial-oriented papers, some technologies have been discussed and proposed on how to implement API governance, as well as others that focus on packaging and cataloging of APIs in a commercial context [16–18]. Such technologies, often referred to as API management platforms, enable API publishing and deployment through a network dashboard. Part of the supported features are policy and access control, back-end guard systems through quotas and rate limits, automatic API deployment, and testing. The dashboard provides additional information on API status, as well as script automation to integrate API deployment into Continuous Integration (CI/CD) pipelines [19]. These technologies can certainly offer valuable services in terms of packaging and cataloging. However, there is a tight coupling between these technologies and API implementation and deployment. Not only that, these systems do not provide stewardship on all aspects and phases of API governance such as combined policy, implementation, and deployment control of APIs for IT-managed services and digital assets [18].

Krintz et al. [16] put forward a strategy on how to improve features present in API development and management, such as API design, documentation, and tools. The authors claim that software development is transitioning from a constant stream of new creations to a reuse-based approach, and APIs play a key role in this change. APIs are generally seen as a high-level resource, providing access to a vast range of functionality. Different APIs provide different abstraction levels of functionality, at different levels of difficulty. As such, they enable programmers to develop on top of existing work, instead of creating code from scratch. However, this ease of access can create impediments in the development of programs, as the developer is limited by their understanding of how to use the API. Correct API design, development, and management mitigate this issue.

Wolski et al. [4] claim that with the emergence of cloud computing, investment and research resources should be put into digital assets such as code, data, and software environments, rather than infrastructures. High-quality, low-cost IT infrastructures are offered by public clouds. As such, efforts should be turned towards maintenance, protection, and life-cycle control of the digital asset [4]. Cloud computing propelled the concept of digital assets as a "Service". The authors discuss that in a business context, there is an increasing focus on employing APIs as access points to encapsulated web services. These are well defined and network available APIs, that defines what operations can be executed on an asset, by whom and under what circumstances. APIs further enable the decoupling of access functionality implementation from the technologies used to manage the asset. The main benefits of employing API are the control over the implementation, operation and access control to assets, asset life-cycle, and scalability. As such, to provide the value their customers expect, these APIs must be carefully developed and managed. The authors suggest that the focus in research and technology related to API diverts slightly from governance aspects. As a result, the authors focus on methodologies and systems for implementing API governance which attempt to automatically determine API similarities. Criticality of API similarity is grounded on the claim that existing API may be extended or re-factored based on emerging new technologies. Understanding how well the API supports the new technology, the effort needed to port assets to a new API and the extent of the impact it causes are deemed as imperative aspects of change control, and as such, of API governance [4].

Krintz et al. [20] claim that despite the importance of APIs

in the IT field, few advances have yet been done to implement API governance. As previously presented in this section, a few commercial technologies exist to address the issue [5], [20]. The work presented in Krintz et al. [20] further addresses the topic of cloud computing, where cloud computing as-a-service (PaaS) technologies are presented as having an advantageous position to aid IT management in their API governance efforts. PaaS is seen as an ideal mean to bring understanding and help to the API governance field. This view is based on PaaS characteristics, such as its inherent distributed nature, scalability, fault tolerance, and successful track record in automation of configuration, deployment, and monitoring [20]. PaaS systems usually take an API-centric approach, offering high-level government of the abstractions they implement, as well as the deployed software. This API-centric approach simplifies the implementation of API governance solutions, by decoupling implementation of digital asset access, and the technology that is in place to store and manage the asset itself. The authors propose methodologies and systems for API governance, focusing on several API management aspects: cataloging, search, and deployment support. Furthermore, special emphasis is put on the following aspects.

- Change control: when changes are required, the side effects shall be predictable and executed in a consistent method.
- Policy Specification and Analysis: only authorized clients shall access resources. API governance should demand development of access control policies.
- Consistent Policy Implementation: policies that control the use of assets shall be implemented consistently independent of the technologies that are used to implement the assets themselves.
- Implementation Portability: as technology changes, API integrity must be preserved over implementations.
- Monitoring and Auditing: API governance must incorporate a unified method to monitor and audit API activity.

The above-listed aspects are deemed as relevant additions to API governance. As such, we include them as a central part of governance aspects in the first iteration of the framework. The remaining background work analysis provides the required inspiration for the creation of the first iteration. The outcome of the findings drawn from background work is subject to evaluation and validation by the five cooperating partners and further iterated upon throughout the duration of this research.

III. Research Methodology

A. Contexts and Case Companies

This thesis is part of a research project conducted at the Software Center [8]. As part of a continued project, five companies invested in API management and strategies cooperated in this research, enabling data elicitation, evaluation and validation. The companies are briefly described in the following.

- Company 1 (***C1***) provides technical solutions, mainly to other companies. The focus is on governance over internal platform APIs and application developers.
- Company 2 (***C2***) provides services and infrastructures. APIs are used in distributed software development.
- Company 3 (***C3***) provides products and services. The APIs are employed in connectivity among devices and related to business process development.
- Company 4 (***C4***) provides services where customers access databases used to generate reports supporting tasks such as quality control. Internal APIs are deployed to service such tasks.
- Company 5 (***C5***) provides electronic products for private and public customers. APIs are used to provide access to their physical devices and cloud.

B. Research approach

This thesis was conducted as a design science research (Fig. 1). Design science is a research-focused search process that aims to create and evaluate artifacts that address important issues [19]. Design science aligns with the main purpose of our research, which is to create a framework for guidance over API governance. We followed an iterative cycle with three iterations based on the design science methodology proposed by Peffers et al. [21] and adapted to the context of this research. In between each iteration, we improved the artifact using the outcome of the previous iteration as input.

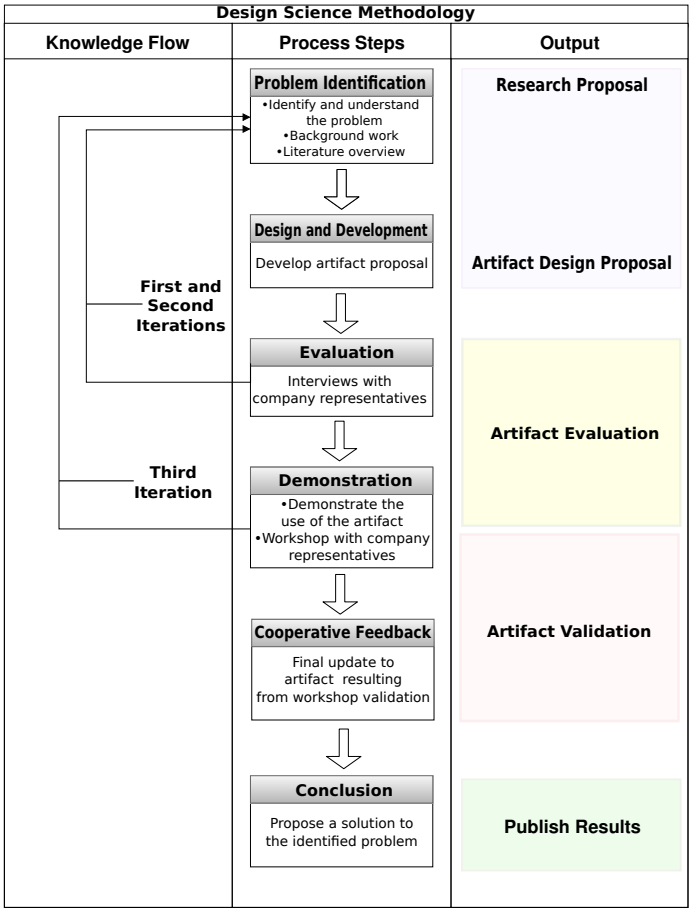

Fig. 1. Design science process use in this research

In their paper, Peffers et al. present, demonstrate, evaluate, and discuss methodology on how to perform design science research, emphasizing its relevance as a discipline focused on the creation of successful artifacts [21]. Adapting from Peffers et al. the methodology used in this research follows six activities:

- **Problem identification and motivation.** This activity relates to the study of the research problem and knowledge acquisition, as to define the problem domain and motivate the value of the solution. This process involves literature review and an analysis of the background work of the project the thesis is inserted in.
- **Design and Development.** The artifact is developed in an iterative manner, based on the findings up to this point.
- **Evaluation.** This activity assesses how well the artifact addresses the solution to the problem. The objectives of the solution are, in the context of this research, assessed by collecting qualitative data through interviews. A comparison between the result of literature findings and the collected data supports the development of the artifact, enabling possible improvements. As such, this activity allows addressing the structure and correctness of the artifact.
- **Demonstration.** This activity relates to demonstrating the use of the artifact as a solution to the identified problem. In the context of this research, the demonstration phase was carried out through a workshop with group focus.
- **Cooperative Feedback.** This activity uses the outcome of the workshop as input to produce the final updates to the artifact. Additionally, the research paper is concluded.
- **Conclusion.** In this activity, a solution proposal in the form of the final artifact and research paper are published.

Interviews: A total of five interviews were conducted with six different individuals from the five cooperating partners. C2 was represented by two individuals, while the others were single individual interviews. The interviews lasted approximately 30 minutes. Interviews were semi-structured with various open-ended questions, promoting non-anticipated answers and providing qualitative data [22]. A list of predefined questions (see Appendix A1 and B1) was created, however during the interviews and when relevant, spontaneous questions were asked, as well as follow up unscripted questions which allowed us to explore further details beyond the provided answer. We aimed to construct unbiased questions as to mitigate the risk of influencing the interviewee's answers. Responders represent different roles and stakeholder groups. The contact with the responders was enabled by Software Center researchers.

The interviews were recorded for later transcription, interpretation, and analysis through a thematic method [23]. Identified themes in the collected data are organized by employing a coding method to discover patterns [24]. Coding facilitates the understanding of the answers and received feedback on the artifact. Each interview was transcribed and coded by a single researcher. The transcripts were reviewed by the researcher who did not do the transcript in order to ensure correctness. Once this process was concluded, both researchers read through the coding together discussing the attributed coding label, what relevant information to extract from it, and what information is not relevant to the API governance topic.

Next, we provide a description of the research iterations and what activities were therein carried out. The iterations followed the design science methodology presented above. Table I presents a summary of each research iteration detailing the utilized data collection methods and participating partners.

TABLE I
ITERATIONS DATA COLLECTION AND PARTICIPANTS OVERVIEW.

Iteration	Data Collection Method	Companies
First	Three interviews	C1, C2, C3
Second	Two interviews	C4, C5
Third	Workshop	C1 - C5

C. First and Second Research Iterations

In the first iteration, we defined the research problem and acquired knowledge about it, to define the problem domain and motivate the value of the solution. Knowledge acquisition included related literature review and analysis of the background work done in the Software Center project. Literature review provided the basis for an initial proposal of relevant aspects and strategies. The research papers and data collected as part of the Software Center project enabled the understanding of how participating partners use API as part of their business strategy, the current status quo in terms of API governance, and an initial perception of the importance of establishing a team responsible for steering the governance process forward. Based on our findings, an initial version of the artifact was produced. Following the artifact creation, three interviews with representatives from the companies with an ongoing governance process were held, corresponding to the evaluation activity. Interviews enabled the collection of data on how governance is carried out in the context of the participant's organization (see Appendix A1). The first iteration of the artifact based on literature findings and background work was presented to the participants. Feedback was received addressing positive aspects, potential improvements, and particular points that we should reconsider. As a result, the feedback was used as input in the next iteration, enabling the framework to be updated accordingly into its second iteration.

In the second iteration, two interviews were conducted: one company without a formal governance process, and another with ongoing changes to their governance process. The interviews were conducted in the same semi-structured manner (see Appendix B1). The focus of the interviews was to understand whether there is a plan in motion to implement an API governance process, and what goals and needs of the organization are to be satisfied.

This interaction allowed for new perspectives on governance from a point of view where a concrete process is yet to be implemented, as well as evaluation of the artifact and the improvements done from the input resulting from the first iteration. The outcome of these interviews is then utilized as input to update the artifact to its third version, leading into the third and final research iteration.

D. Third Research Iteration: Workshop

In the final iteration, we demonstrated the artifact on a cross company workshop with group focus. Workshops offer the means to achieve a goal, where a group of people is gathered to promote learning, acquiring knowledge, or discuss a domain specific issue [25]. The workshop's goal was to measure how well the artifact addresses the solution to the problem, by demonstrating the artifact resulting from the previous iterations. Presenting the framework to a group of people as opposed to single individuals enables an environment where constructive discussion is promoted, allowing for the collection of feedback that is a sum of a collective collaboration.

However, the participation turnout was less than the initially planned. The framework was nonetheless presented to attendees and qualitative data was collected in terms of framework relevancy, perceivable trade-offs, and whether they would instantiate the framework or parts of it in their organization's context. As a complement to the workshop, the absent partners were provided with the presentation material and a follow-up questionnaire containing the same questions posed during the workshop (see Appendix C1). A total of 16 people were contacted, resulting in 3 questionnaire answers.

The resulting input from the workshop and complementary questions is utilized as input to make the last updates to the framework, before presenting it as the proposed solution to the identified problem.

Finally, through this thesis report, we communicate the problem and its relevance, exposing the artifact, its value, the correctness of its design and perceived effectiveness.

E. Framework Design Method

1) ***Framework First Iteration****:* Starts by conducting a deep analysis of background and related work within the topic of API governance. Due to the scarcity of research on the topic, the literature review scope is broadened to include API governance in software development, and API design and management. As a result of this analysis, the first iteration of the governance framework is designed. This first iteration is composed of a set of important aspects and strategies to consider during the API governance process (see Appendix A2). Additionally, the framework proposes the creation of a governance team composed of diverse key roles such as developers, testers and software architects. Key participating roles are also extended to the organization, aiming to promote an involvement of all the interested parties of the organizational hierarchy.

2) ***Framework Second Iteration****:* Utilizes the outcome of the first iteration as input. The framework resulting from background work is presented to interviewees, enabling its evaluation and validation. Coding labels are designed as shown in Table II and applied to the transcripts resulting from interviews (see Appendix A3). This approach allows for similarities between the framework and the real-life scenarios at the companies to be found, and to identify missing relevant information.

TABLE II
LABELS USED FOR DECODING THIS ITERATION'S TRANSCRIPTS.

Label	Reasoning
Aspects	Allows the identification of important characteristics.
Strategies	Allows the identification of the planned action.
Challenges	Allows the identification of potential hinders or problems.
Others	Useful information that can be used for further discussions.

The second iteration of the framework is the end-result of the first round of interviews (see Appendix B2).

The framework presented at the first round of interviews is seen as non-agile and overweight when taken in its entirety. This resulted in proposing a new approach to framework instantiation, whereby the implementation of the framework in an iterative manner should be tailored according to the organization's needs. In the second iteration, we propose that the company picks and chooses what aspects and strategies best suit their needs for each iteration, as opposed to implementing the entire framework all at once. Furthermore, governance board replaces governance team, introducing more than a simple name change: the responsibilities of the board are revised, and members' composition are defined with clearer roles. Lastly, organization key roles specification is removed in this iteration. Input from the first iteration revealed that it was too reliant on a hierarchical organization, deemed as too narrow of a view, as flat structured organizations could not map most of the roles. The update focuses on promoting the idea that the organization as a whole should be aware of the governance process, supporting and promoting the governance board to make decisions.

3) ***Framework Third Iteration****:* Uses the previous iteration as input. The framework is updated and presented in two interviews. The transcripts were labeled as shown in Table III. This iteration sees the introduction of new labels as the interviews adopt a higher focus on validation and evaluation of the artifact (see Appendix B3).

The third iteration of the framework is the outcome of the second round of interviews (see Appendix C2). Notable changes are a revision of the description of the listed aspects and strategies and improvements made to traceability of the components of the framework. Each component is supported by and mapped to background work and companies. The newly introduced instantiation and implementation guidance, as well as Governance Board are endorsed by both interviewees.

TABLE III
LABELS USED FOR DECODING THIS ITERATION'S TRANSCRIPTS.

Label	Reasoning
Aspects	Allows the identification of important characteristics.
Strategies	Allows the identification of the planned action.
Challenges	Allows the identification of potential hinders or problems.
Governance board	Allows the identification of potential changes to the composition and responsibilities of the board.
Aspects evaluation	Provide qualitative data on governance aspects.
Strategies evaluation	Provide qualitative data on governance strategies.
Framework instantiation	Provide qualitative data on framework instantiation process.
Others	Useful information that can be used for further discussions.

*4) **Framework Fourth Iteration**:* Takes place following the workshop. The focus is on evaluation and validation of the third iteration of the artifact. A theoretical demonstration of artifact implementation is carried out by promoting the discussion on whether the participating partners present at the workshop would apply the framework, what aspects and strategies they would focus on, and how they would implement it. Furthermore, this iteration sees the introduction of two new models created using input from previous iterations, focusing on perceived challenges relating to decision making processes and framework implementation. These models seek to aid in deciding what strategies and aspects to instantiate at each iteration of framework implementation, as well as in discerning what changes the API governance process should be concerned with, based upon the scope of said change.

An unfortunate setback during this iteration was caused by a less than desired attendance to the workshop. Complementary questions were sent via an online form to the cooperating partners. However, here the answer rate is also lackluster. Notwithstanding the unfortunate events, the received input (see Appendix C3) is analyzed and utilized as input for the final framework iteration and a framework instantiation proposal, both presented in the Section IV.

The received input was generally positive, with specific aspects and strategies emphasized and motivated through the different organizations' needs. Governance board and the newly introduced decision models are further discussion points, receiving the participants' endorsement. Finally, potential framework disadvantages are discussed, as well as whether participants would apply, not apply, or partially apply the framework (see Appendix C3).

IV. FRAMEWORK FOR API GOVERNANCE

This section describes our proposed framework for guidance over API governance.

A. Framework Overview

When dealing with change, the goal and vision shall be well defined and advertised to the organization [26]. Organization higher roles and stakeholders should be aware of the governance process. Additionally, it is important to identify and remove obstacles to change, and people's mindset should be aligned with the changes [26]. As such, and in order to avoid resistance, the support of the organization is crucial for the governance process. It is also critical to understand how people perceive the governance process. Lewin's concept of change details that in order to solve social conflicts, learning obstacles should be removed and individuals should be given their space to understand and rebuild awareness of the world around them [27]. Drawing from this ideal, the framework here presented encourages that interested parties in an API should be educated about the governance process. Individuals should understand why a governance process is necessary, its advantages, and the problem of not having a process.

The presented framework is intended as a complete and all-encompassing API governance process. Therefore, when instantiating the framework, companies should decide upon which aspects and strategies to select based on the company's context, scope, and more importantly, needs. Resulting from the iterative cycles of investigation and validation, the following main characteristics of the API governance framework were identified. These provide an overview of how the governance process should be implemented, and what context it can be applied to.

- **Iterative process:** carried out in iterative cycles where improvements are possible until the final goals are achieved.
- **Re-usability:** the framework does not target any specific context. Instead, it is seen as applicable to the context of different organizations where API governance guidance is necessary. With that said, certain aspects of this framework might not be applicable to certain contexts. As such, it is up to those intending to adapt this framework to identify the context in which the framework will be deployed and use their best judgment to adapt it to that context.
- **Monitoring and traceability:** documentation for traceability provides information about the framework implementation process while keeping the goals in sight. This framework recommends documenting in an agile manner [28]. To document continuously enables the creation of a sufficient document with just the required information to satisfy its purpose.

Fig. 2 provides a complete overview of the framework. Aspects, strategies, and governance board encompass the API during its life-cycle and are implemented gradually over an iterative process comprised of three phases. Next, in this section, we present a detailed description of each component.

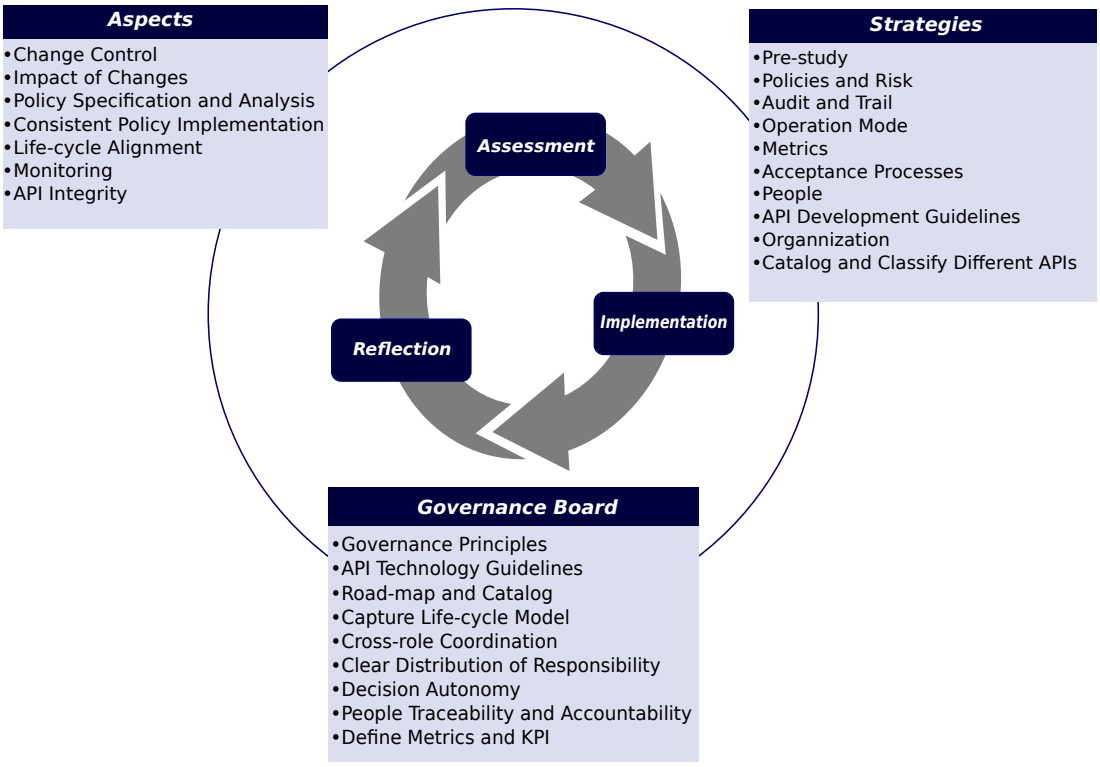

Fig. 2. API Governance Framework Overview.

B. Phases

To deploy the framework, we suggest an iterative approach consisting of three distinct phases as shown in the center of Fig. 2. These phases draw inspiration from the work of Burnes [27], where Lewin's 3-step model, one of the theories for planned change, is discussed. The 3-step model outlines three steps for a successful change effort: *Unfreezing, Moving, Refreezing*. ***Unfreezing*** refers to a disequilibrium in the forces that affect the status quo. ***Moving*** speaks to the process of evaluating options and sets the course for change. The proposed ***Assessment*** phase evaluates the current status quo and employs forces of change in order to move the process forward. ***Implementation*** phase draws from the concept of setting the course for change, whereby the process of implementing the change is conducted. ***Refreezing*** tries to stabilize at a new quasi-stationary equilibrium. ***Reflection*** phase relates to ***Refreezing*** by auditing the new status quo and stabilizing the implemented changes.

1) ***Assessment phase****:* Current status quo. In this initial phase, the focus is on assessing the current status quo of API governance. From the framework overview, relevant aspects and strategies are identified. By identifying the points that the organization is lacking in, it is possible to move forward to the Implementation phase.

2) ***Implementation phase****:* The new status quo. This phase oversees the implementation of the aspects, strategies and participating roles provided in the overview. Personnel is appointed roles based on key skills. Aspects of governance are considered and analyzed and applied to the governance process.

3) ***Reflection phase****:* Auditing the new status quo. This phase concludes the implementation by producing an audit on how API governance is now performed. Traceability is performed through documentation for future reference and other artifacts. These outcomes are available for re-use in later iterations. When the results are deemed as adequate to the organization's context, the iterative process should then be terminated. However, flexibility of the model allows for a re-evaluation of the status quo and re-start of the activities if necessary.

C. Aspects and Strategies

In Tables IV and V we list aspects and strategies resulting from our findings and which could be considered and deployed by an API governance process. Additionally, the column "Supported by "provides traceability to the listed aspects and strategies by mapping to background work research (references) and data collected from collaborating partners during interviews (C1-C5).

TABLE IV
GOVERNANCE ASPECTS.

Aspects	Description	Supported by
Change Control	When changes are required, the resulting effects shall be carefully considered and executed in a consistent method. The long-term vision and purpose of an API should be identified and preserved. The pre-study strategy suggested by this framework helps providing an overview of the expected outcome of a change. Rollbacks to the change should restore functionality also in a consistent and complete manner.	[6], [16] •***C2***,***C4***,***C5***
Impact of Changes	In the context of business, APIs are part of a broad context and any changes to the API can have influence at a business level and IT operations. As such, the impact of changes shall be carefully evaluated. Stakeholders of an API, such as consumers and business owners, shall be informed of changes and what impact they create.	[6], [16] •***C2***,***C4***,***C5***
Policy Specification and Analysis	API governance shall be concerned with access control policies, their analysis and application. Who should access an API, and who should not must be decided taking into account the business context. APIs as access points to assets shall only allow authorized clients access to the asset.	[6], [16] •***C5***
Consistent Policy Implementation	API governance should ensure that APIs are independent of the technologies that is used to implement the assets. Decoupling API from asset implementation allows for API integrity to be kept: changes to one do not influence the other.	[6], [16] •***C2***,***C4***,***C5***
Monitoring	API governance shall be concerned with monitoring API activity. The proliferation of APIs requires new approaches to control and govern APIs.	[6], [16]
Life-cycle Alignment	The governance process is involved in all the duration of the API life-cycle. For instance, through monitoring API activity, if a decision is made to deprecate the API, the governance process shall ensure that it is not to be awoken again.	•***C1***,***C2***
API Integrity	An API shall be able to interface on a newer version of the platform without conflicts, and without effort. When planning new features, existing API should not require extensive refactoring, and backwards compatibility shall be ensured over a period of time. As an interviewee expressed, API governance ensures that "people do not do their own probably incompatible change between other changes".	•***C1***,***C3***

TABLE V
GOVERNANCE STRATEGIES.

Strategies	Description	Supported by
Pre-study	allows to foresee consequences of requests. A thorough analyses shall be conducted in order to understand the impacts of a change request to the API. The outcome from the analyses is then used to assist the governance board with its final decision.	•***C2***
Policies and risk	relate to the correct method to work. These strategies are also connected to risk control. Policy strategies ensure that there is compliance with laws, regulations, security control, risk mitigation strategies, corporate guidelines and industry best practices.	[17] •***C3***
Audit and trail	for instance, via version control systems enables tracking and management of changes to an API.	***C2***
Operation mode	establishes a set of goals to guide the development of the API governance process and enable the collection of metrics. For instance, establish what scale of changes the API governance process should be concerned with (small scale changes versus large scale changes).	[17]
Metrics	are the means to provide visibility to governance. The following are suggested metrics to consider: time for changes to be approved, number of approvals and refusals, stakeholder satisfaction scores, technical debt resulting from accepted changes, business impact resulting from accepted changes and service downtime caused by changes. These metrics can be contrasted with the determined policies and operation mode, allowing for an understanding and evaluation of the governance process.	[17]
Acceptance processes	provide the chance to develop and test an API and its compliance with the policies, as well as to make a decision on whether it should be accepted or not. These processes can be automated or human controlled.	•***C1***
People	should be supported and encouraged to make decisions and requests regarding API changes and features. Who can accept changes, who can request changes, and who can implement changes are all equally important aspects to consider.	[17] •***C4***
API development guidelines	ensures that APIs are built up to a certain standard. Thoroughly documenting these guidelines ensures that APIs are built in a consistent manner, and can be worked on (developed, maintained) by different people. Additionally, business related roles such as product owners can have a better view of the development process by consulting these documents.	[16], [17] •***C5***
Organization	is responsible for nurturing a culture of support and reward for good governance over the APIs.	[17] •***C1***
Catalog and classify	different APIs, providing an accessible method of grouping APIs, for instance with documentation. API versioning could provide a unique identifier to unique states of an API, allowing for clear cataloging and classification.	[4], [16] •***C2***

D. API Governance Board

In this section we suggest the creation of an API governance board, with the responsibility to monitor and manage the implementation of the framework and API governance. The proposal for the establishment of a governance board is heavily influenced by previous Software Center background work and endorsed by the cooperating partners during interviews and workshop. From the Software Center background work, it was perceivable that some of the five cooperating partners already employed a tactic similar to what is here proposed in the form of a governance board and corresponding responsibilities. The board should operate as interface guardians, acting to protect the API integrity at all times. Additionally, the board should be at the helm of the governance process throughout the API life-cycle: from design, creation, testing, utilization, management, and retirement.

The core of the team shall be composed of people who understand the "why and the what", and those who understand the "how". The "why and what" helps maintain the balance enabled by members capable of a broader vision of the entire system. Key participating "why and what" roles could be Software Architects, System Architects, Domain Experts or similar roles. The "how" contributors provide understanding of how technology is implemented. Key participating "how" roles could be for instance Lead Developer or similar technical roles. Both sides should come to a common agreement before moving forward with decisions. However, it is not discarded that an individual might have both knowledge of "why and what" and "how" and that way fulfilling the required knowledge required to compose the governance board. As emphasized during the interviews with the participating partners, this balance act is vital for the API governance process as both dimensions are equally important in order to maintain API integrity.

The suggested framework, when fully implemented, can certainly be perceived as heavyweight and produce a time-consuming API governance process, as seen in the feedback received from company 4. As such, we suggest the governance board to determine whether a change request should trigger the whole governance process, or instead if the request should be sent into the backlog of the team that is to implement the change and prioritized internally and accordingly. By employing the pre-study strategy and analyzing the change its impact, the governance board members should be able to determine the scope of the request and decide how to best proceed.

The governance board is of paramount importance to ensure that the API governance process is carried out, but also to see that the process is compliant with organization values and policies. The API governance board should be empowered to address the following responsibilities present in Table VI.

TABLE VI
API GOVERNANCE BOARD RESPONSIBILITIES

Responsibilities	Supported by
Governance principles. Governance board should be concerned about what principles to follow to ensure the effectiveness of the governance process. Regular assessment of compliance with policies and operation mode strategy defined by the organization should be conducted.	• [17] •Previous Software Center Elicitation •Endorsed by cooperating partners
API technology guidelines. Create and maintain guidelines for API technology and development. The governance board should be enabled to make suggestions and decisions on what technology to use in API development, securing API integrity and business value.	•Previous Software Center Elicitation •Endorsed by cooperating partners
Road-map and catalog. Establish a road-map with a well-defined time-frame and goals, and update it regularly, enabling an execution plan with milestones according to the road-map. These could be goals related to pre-study completion, impact of change analysis or establishing an acceptance process for API changes. When building the road-map stakeholder input should be added in order to identify goals that provide business value.	• [17] •Previous Software Center Elicitation •Endorsed by cooperating partners
Capture life-cycle model. Create a model that captures the life-cycle of the API, for internal and external API users. If the API is deprecated, the governance board ensures that the API is not awoken again.	• [17] •Previous Software Center Elicitation •Endorsed by cooperating partners
Cross-role coordination. Ensure that communication channels between the participating roles are kept open to all the interested stakeholders.	• [17] •Previous Software Center Elicitation •Endorsed by cooperating partners
Clear distribution of responsibility. Clearly map roles to responsibilities so that everyone in the governance board is fully aware of what they should really do, instead of what they believe they do.	• [17], [29] •Endorsed by cooperating partners
Decision autonomy. Organization should allow the governance board to make important decisions related to API governance. This approach helps avoiding potential bottlenecks by removing the need to constantly report to a single person or "higher ups" from the organization.	• [17], [29] •Endorsed by cooperating partners
People traceability and accountability. Associate people to their respective responsibilities. The concept of "who did what", enables a go-to-person to consult when the understanding of some change or topic is necessary.	• [17], [29] •Endorsed by cooperating partners
Define metrics and KPI. Quantifying business value and metrics increases progress and process visibility to the organization. These are tied to the Metrics strategy.	• [17], [29] •Endorsed by cooperating partners

E. Framework Decision Models

Different organizations have different needs and the framework here proposed should allow flexibility. It is not expected that the framework shall be instantiated all at once. Instead,

the *Assessment Phase* (refer to Section IV-B1) shall be used to identify the aspects and strategies that better answer the needs of the organization. For instance, depending on which of the presented strategies are implemented, overhead to production can be introduced, increasing time to delivery. Furthermore, our findings reveal that it is arguable whether each and every change request should go through the entirety of the API governance process. The scope of change could play a role in this assessment. Bottlenecks can be prevented by ensuring that small scale change requests do not hinder other more impactful change requests from being governed. In order to assist in this decision-making process, we suggest models based in the cost-value models present in [30], [31]. These decision models are intended to be easy to use and provide help, ensuring that the models can be used in software development [31]. The models were adapted to fit the given framework context and the comparison values are derived from our findings.

1) ***Implementation Decision Model****:* The model presented in Fig. 3 helps identifying which aspects and strategies to instantiate. The main concept is to compare the effort that would take to implement a given aspect/strategy versus the value it would bring to the organization. This model allows one to group aspects and strategies to be implemented in a prioritized list. The model is divided in sections: ***A*** representing what is deemed as most valuable to implement right away, ***B*** should be carefully discussed among stakeholders, because it brings high value but takes considerable effort, ***C*** refers to non-crucial things that might not be prioritized, and ***D*** are red flags that point to aspects and strategies that do not add enough value to the current process considering the effort they take to implement. Example of usage of this model is explained in section IV-E3.

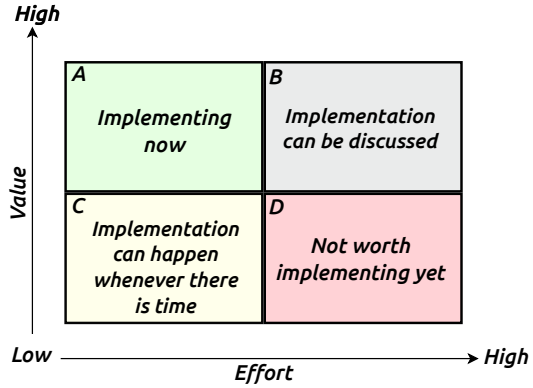

Fig. 3. Implementation decision model.

2) ***Change Decision Model****:* The model presented in Fig. 4 helps to identify which changes the API governance process should be concern with. Similar to model presented in Fig. 3, this model results in a prioritized list. A comparison is made between the scope of a change against its impact into the system. The result can be grouped in different sections. Section ***A*** refers to changes that the governance process should monitor closely, section ***B*** should require definitive attention and monitoring, section ***C*** are changes that could be ignored as to avoid overwhelming the governance board, and section ***D*** are small scope changes to be discussed only if monitoring is necessary or attention is required. Example of usage of this model is later explained in section IV-E3.

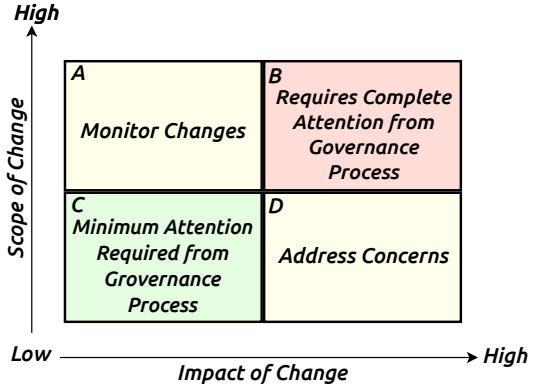

Fig. 4. Change decision model.

3) ***Models Usage****:* As shown in Fig. 5, these decisions are encouraged to be taken by gathering all the API stakeholders together. Once together, discussions can be initiated such as on how to prioritize the presented aspects and strategies according to the organization needs or which changes should the governance process be concerned about.

The following are examples of usage of the proposed models. The prioritization is done by drawing from the perceived context at Company 4, chosen opportunistically due to the discussion triggered during the interview.

- In the case of aspects and strategies, ***metrics*** is deemed as important. However, it is a strategy that requires a long time to implement. Data needs to be collected over a period of time and further analyzed to provide value. This is perceived as requiring a high amount of effort. ***People*** strategies, on the other hand, may bring immediate high value to the organization. This strategy can be effortlessly and quickly implemented, by enabling people to make decisions that could otherwise introduce bottlenecks.
- Deciding which changes the API governance process should be concerned about. ***Internal change request*** between developer teams with no impact to the business value of the API, such as updating the version of a library used in the implementation. These changes can be deemed as requiring the minimum attention of the governance process, as the perceived impact is not extensive, as well having a small scope of change. On the other hand, ***external changes*** originating from external users that request modifications to the core functionality of the API are deemed as having great impact in the system. The scope of the change is considered to be extensive as the changes affect the core functionality. As such, changes of this nature should require full attention of

the governance process "(...) Some (APIs) are internal, others more public. I guess it will be more important to the public to actually have more procedures to follow and be governed".

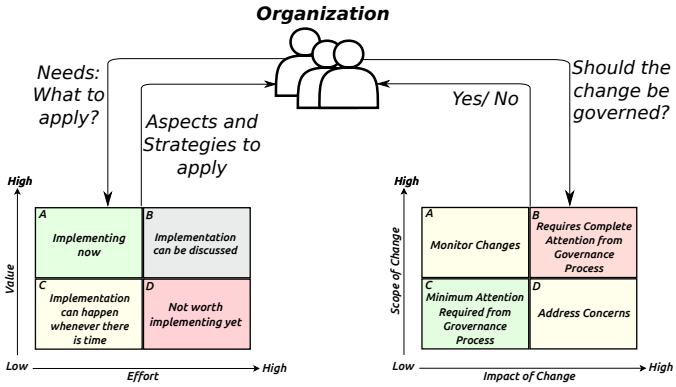

Fig. 5. Decision models' usage.

F. Framework Instantiation

This section discusses framework instantiation within the context of each of the participating partners, drawing from the data gathered during interviews, and elaborated using the above proposed decision model for what aspects and strategies to apply. First iteration data can be found in Appendix A3 and second iteration in Appendix B3.

The duration of the iterations for framework implementation presents the biggest challenge to the instantiation process. The iterations could happen rapidly or slowly, determined by the context of the organization. However, care must be taken to ensure that the process is not too cumbersome or too resource intensive.

Listed are a suggestion of which aspects, strategies and governance board tasks each organization could potentially benefit from. These suggestions are drawn from the qualitative data collected during the interviews and background work from the Software Center project. Furthermore, the establishment of a governance board is proposed for all participating partners. Two case companies report as already having a governance board in place. However, this research details member composition and responsibilities that can bring further value. The governance board proposed by this framework enables the instantiation of a formal governance process, by attributing defined roles, mapping responsibilities to these roles, promoting the establishment of the proposed guidelines and ensuring these are employed.

1) Company One: Table VII presents an instantiation proposal of the framework at company 1. The status quo of API governance process is in operation, with focus on two distinct and equally important areas of governance.

TABLE VII
FRAMEWORK INSTANTIATION AT COMPANY 1.

Topic	**Company 1**
API Governance Maturity	In Operation
Challenges	• Speed of development. • Support different development speeds. • Separation between platform and applications. • Delayed value. • Documentation.
Suggested Aspects	• Impact of changes. • Consistent Policy Implementation. • Life-Cycle alignment.
Suggested Strategies	• Pre-study. • Operation Mode. • Catalog and classify different APIs. • API development guidelines.
Governance Board Considerations	• Governance principles. • Governance principles. • Road-map and catalog. • Capture Life-cycle model

2) Company Two: Table VIII presents an instantiation proposal of the framework at company 2. The status quo of API governance process is of in full operation.

TABLE VIII
FRAMEWORK INSTANTIATION AT COMPANY 2.

Topic	**Company 2**
API Governance Maturity	In Operation
Challenges	• Compatibility with different technologies. • Cross-component feature development. • Resources for features and update. • Parallel work. • Technical debt.
Suggested Aspects	• Impact of changes. • Monitoring.
Suggested Strategies	• Policies and risk. • Audit and trail.
Governance Board Considerations	• Establish a Governance Board. • Governance principles. • Cross-role coordination. • People traceability and accountability.

3) Company Three: Table IX presents an instantiation proposal of the framework at company 3. The status quo of API governance process is in operation.

TABLE IX
FRAMEWORK INSTANTIATION AT COMPANY 3.

Topic	**Company 3**
API Governance Maturity	In Operation
Challenges	• Faster and improved development. • Make use of established solutions. • Enable data analytics. • People as decision makers can create bottlenecks. • Different approaches to governance process depending on departments
Suggested Aspects	• Change control. • Monitoring. • Life-Cycle alignment.
Suggested Strategies	• Pre-study. • Policies and risk. • People. • Catalog and classify different APIs.
Governance Board Considerations	• Establish a Governance Board. • Governance principles. • Decision Autonomy.

4) Company Four: Table X provides an overview of framework instantiation at company 4. The interview with this partner revealed a stage of informal procedures in terms of API governance. The status quo was that of brainstorming, planning, and uncertainty.

TABLE X
FRAMEWORK INSTANTIATION AT COMPANY 4.

Topic	**Company 4**
API Governance Maturity	Planning
Challenges	• Improve service tasks. • Coordination between developer teams. • API governance only tried at an informal level.
Suggested Aspects	• Change control. • Impact of changes. • Life-Cycle alignment.
Suggested Strategies	• Pre-study. • Acceptance Process. • People. • API development guidelines.
Governance Board Considerations	• Establish a Governance Board. • Capture life-cycle model. • Governance principles.

5) Company Five: Table XI provides an overview of framework instantiation at company 5. The status quo of API governance process is in operation yet there are plans to undergo changes.

TABLE XI
FRAMEWORK INSTANTIATION AT COMPANY 5.

Topic	**Company 5**
API Governance Maturity	In Operation with Planned Changes
Challenges	• Provide enriched content while maintaining trust and customer connections. • Collect customer data. • Struggle with impact of changes • Short time to market. • Change existing APIs may harm business.
Suggested Aspects	• Impact of changes. • API Integrity. • Consistent Policy Implementation.
Suggested Strategies	• Pre-study. • Policies and risk. • Operation Mode.
Governance Board Considerations	• Establish a Governance Board. • Governance principles. • Capture Life-cycle model.

V. DISCUSSION

We have presented the relevant aspects (**RQ.1**) and strategies (**RQ.2**) for guidance over API governance. Furthermore, we captured and presented these aspects and strategies in a conceptual framework (**RQ.3**). The framework is proposed as a solution to the identified problem of guidance over API governance. The results are an outcome of data collected throughout this study from background work, collaborative research with several large companies, and an iterative refinement process of the presented artifact. We found which aspects and strategies are relevant in API governance, bringing value to the companies and helping achieve their business goals.

Our work differs from the identified and reviewed related literature, whereby scientific studies tend to focus on governance over software development in general or, when concerned with APIs, emphasize API governance over particular fields (e.g. could computing). We contribute to the API governance field with a result which we argue to be new, useful and applicable.

A. Artifact Development

The framework evolved past its initial envisioned content of relevant aspects, strategies and their trade-offs, to include implementation strategies and the governance board. Adopting a framework, or parts of it, means undergoing change. As previously discussed, change can be met with resistance, pointing to the criticality of carefully planning the change effort. Drawing from agile and change management methodologies, we included in the framework an iterative implementation strategy as a solution, in order to expedite the change process. This approach was well received by the cooperating partners, “it feels kind of natural...how we work”, stated one of the interviewees. Care must be taken to ensure the implementation process does not demand too many resources or creates too much of a burden on the organization. We do not report on how

long the duration of iterations and phases of the implementation process should take. We believe that iteration, phases, or both, could happen rapidly or slowly depending heavily on the organization's context. The conceptual framework may be implemented differently determined by company's needs, picking and choosing from the proposed aspects and strategies with a calculated approach.

1) ***Governance Board****:* Governance board is included as a component of the framework. Governance relates to not only aspects and strategies, but also on the people who sustain the governance process. Our discoveries point to people playing a key role in a governance process. Often, acceptance processes for changes over API are dependent on a single person, creating a potential bottleneck. Furthermore, without the proper expertise, people can make misinformed decisions. The governance board draws inspiration from companies with an in-operation governance process who possess a similar team (**C1,C2**), and seeks to tackle the mentioned issues.

2) ***Aspects****: Change Control* and *Impact of Changes* are often a challenge (**C2,C4,C5**) and should thus be a focus. Throughout the interviews, a discussion sparked on the topic of detaching API governance from API implementation details. Meaning, API governance should not be concerned with detail implementation of an API, such as technology choice, development guidelines or code. Our findings point to most of the companies preferring to include people with expertise in API development as part of the governance process (**C2,C4,C5**) "keeping it totally separated(...) leads to governing and ID rather than what is being implemented.". Finally, *Life-cycle Alignment* and *API Integrity* originate from interviews. While these are perceivable in the background work exploration, a formal inclusion as relevant aspects are supported by the focus that collaborating partners put on supporting an API from its inception to retirement, as well as protecting business value by safeguarding API integrity.

3) ***Strategies****:* The participating partners expressed interest and positively evaluated most of the proposed strategies. We observed that *People*, *Organization* and *Policies and Risk* were the initially presented strategies that fit in a broader context. Resulting from interviews, *Pre-study* and *Acceptance Processes* were introduced. Resulting from the first iteration, *Pre-Study* resulted from input from **C2**, where the practice is in place. Additionally, *Acceptance Processes* is a re-factored strategy from the initial proposal that seeks to better address challenges identified in **C2** and **C3**. These are valuable additions and were well received during the following iterations, becoming in our eyes ideal candidate strategies to use when implementing the framework for the first time.

However, there was a perceivable resistance to certain strategies. Often connected with documentation, strategies such as *Audit and Trail*, *Metrics* and *Catalog and Classify Different APIs* were deemed as valuable yet resource intensive. With that said, most companies (**C1,C2,C4,C5**) described challenges related to documenting and auditing APIs, keeping track of changes and parallel work. These strategies produce clear benefits, but at the cost of resources which companies are often unwilling to spend. Time to market, market competition, and customer value dictate that companies forgo these strategies. We argue that the value versus effort trade-off of these strategies should not be underrated. We believe these should be regarded as highly valuable strategies to be implemented in later iterations, instead of the most immediate ones to focus on when instantiating the framework.

4) ***Workshop****:* Due to time constraints, it was not possible to instantiate the framework in a real-life scenario at a company. The workshop was indented to demonstrate the use of the framework by engaging the companies in open discussion about the framework, motivating how they would approach its instantiation and what parts of the framework they would initially focus. The input from the workshop continued the trend of previous iterations regarding validation of the identified aspects, strategies, and governance board. Responders expressed concerns related to overhead and difficulty in reaching agreements. We believe the proposed decision models can help in mitigating these issues. Regarding business strategies, we believe these to be outside of the scope of this research, yet it could certainly be an interesting topic for future work.

B. Future work

Adding to business strategies in API governance, it could be interesting to follow and study the implementation of the framework in a real-life scenario over an extended period of time, seeking further validation of the framework. Furthermore, focus could be put towards detailing the listed aspects and strategies as to provide a more in-depth step-by-step guide to API governance. Finally, one of the main questions raised during this research is whether every change should go through the governance process. This could prove beneficial to investigate with further detail in an attempt to reduce potential bottlenecks introduced by a lengthy governance process.

VI. Threats to Validity

The reliability of research is tightly coupled with the validity of its results [32]. The principal threats to the validity of this research were identified and are presented in this section.

A. Construct Validity

Construct validity refers to whether the measuring of the construct measures what it claims to measure [33]. In this study, inexperience from the researchers in conducting research may have led to sub-optimal question formulation and workshop preparation. Interview scripts were prepared entirely by the researchers. To mitigate this threat, the script was presented to the thesis' supervisor for feedback and revision.

B. External Validity

External validity relates to which degree the research findings can be generalized to other studies [33]. This research can be easily reproduced by others as all important steps are

documented in section III. During this research we denoted that each of the five different organizations had different contexts, needs, and challenges. We believe that conducting a similar research with new organizations could potentially result in the introduction of new components of API governance, which would require evaluation and validation. However, the proposed framework is composed of components which are commonly agreed upon and were positively received among the organizations, leading us to believe it can be generalized to companies that use APIs as a service providing layer.

C. Internal Validity

Interpretative validity is concerned with what the object of study means to those engaged in and with the object [32]. Coded transcriptions from recordings imply a degree of interpretation by the researchers, which poses a reliability threat. When coding transcripts, the goal is to understand a phenomenon from the point of view of the participants and provide an interpretation of statements. To mitigate this risk, the transcript analysis followed a thematic method [23]. Additionally, both researchers discussed thoroughly each coding label attributed to transcribed statements until an agreement was reached.

VII. Conclusion

Our work seeks to contribute to the identified gap in research on API governance. We have presented a conceptual framework for guidance over API governance. The framework poses as the solution to the identified problem and is built from background work and collaboration with several large-scale companies. With the framework, we provide valuable insight on what aspects and strategies to consider for API governance. Despite the reported challenges in demonstrating the usage of the framework, the collected data from participating partners validates the framework as useful for organizations concerned with API governance. Drawbacks are identified relating to a perceivable heavy-weight nature of the framework, however, we propose solutions to enable a light-weight instantiation.

We see our results as promising, leading us to believe that further framework validation through implementation on a real-life scenario is worthy of future work to investigate guidance over API governance.

Acknowledgment

We would like to thank our supervisor Jennifer Horkoff for proposing this topic, and for her guidance and support. Additionally, we would like to extend our deepest gratitude to all cooperating partners for their availability and valuable input.

REFERENCES

[1] J. Bastos, L. Afonso, and C. de Souza, "Metacommunication between programmers through an application programming interface: A semiotic analysis of date and time APIs," *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2017.

[2] C. de Souza, D. Redmiles, L. Cheng, D. Millen, and J. Patterson, "How a good software practice thwarts collaboration," *ACM SIGSOFT Software Engineering Notes*, vol. 29, no. 6, p. 221, 2004.

[3] I. Hammouda, E. Knauss, and L. Costantini, "Continuous API Design for Software Ecosystems," in *2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering*, May 2015, pp. 30–33.

[4] R. Wolski, C. Krintz, H. Jayathilaka, S. Dimopoulos, and A. Pucher, "Developing Systems for API Governance," Sept 2013. [Online]. Available: https://figshare.com/articles/Developing_Systems_for_API_Governance/790746

[5] A. de O. Luna, P. Kruchten, and H. de Moura, "Agile Governance Theory: conceptual development," *CoRR*, vol. abs/1505.06701, 2015. [Online]. Available: http://arxiv.org/abs/1505.06701

[6] H. Jayathilaka, C. Krintz, and R. Wolski, "EAGER: Deployment-Time API Governance for Modern PaaS Clouds," in *2015 IEEE International Conference on Cloud Engineering*, March 2015, pp. 275–278.

[7] C. Krintz and R. Wolski, "Unified API Governance in the New API Economy," Cutter Consortium, 37 Broadway, Suite 1 Arlington, MA 02474, U.S.A., Tech. Rep., Sept 2013.

[8] "Software Center." [Online]. Available: https://www.software-center.se/research-themes/technology-themes/customer-data- and-ecosystem-driven-development/api-strategies/

[9] J. Horkoff, J. Lindman, I. Hammouda, and E. Knauss, "Experiences Applying e3 Value Modeling in a Cross-Company Study," *ER*.

[10] J. Horkoff, J. Lindman, I. Hammouda, E. Knauss, , J. Debbiche, M. Freiholtz, P. Liao, S. Mensah, and A. Strömberg, "Modeling Support for Strategic API Planning and Analysis," *ICSOB*.

[11] J. Lindman, I. Hammouda, J. Horkoff, and E. Knauss, "Emerging Perspectives to API Strategy," *IEEE software*.

[12] P. Weill and J. Ross, *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Boston, MA, USA: Harvard Business School Press, 2004.

[13] P. L. Bannerman, "Software development governance: A meta-management perspective," in *2009 ICSE Workshop on Software Development Governance*, May 2009, pp. 3–8.

[14] F. Haupt, F. Leymann, and K. Vukojevic-Haupt, "API governance support through the structural analysis of REST APIs," in *Computer Science - Research and Development*, 2017. [Online]. Available: https://doi.org/10.1007/s00450-017-0384-1

[15] J. Stylos and B. Myers, "Mapping the Space of API Design Decisions," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*, Sept 2007, pp. 50–60.

[16] C. Krintz and R. Wolski, "Unified API Governance in the New API Economy," Cutter Consortium, 37 Broadway, Suite 1 Arlington, MA 02474, U.S.A., Tech. Rep., Sept 2013.

[17] "SOA and API Convergence Strategy and Tactics," WSO2, 787 Castro Street, Mountain View, CA 94041, U.S.A., Tech. Rep., 2014.

[18] ca technologies, "API Management." [Online]. Available: https://www.layer7tech.com

[19] S. Adikari, C. McDonald, and J. Campbell, "A Design Science Framework for Designing and Assessing User Experience," in *Human-Computer Interaction. Design and Development Approaches*, J. A. Jacko, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 25–34.

[20] C. Krintz, H. Jayathilaka, S. Dimopoulos, A. Pucher, R. Wolski, and T. Bultan, "Cloud Platform Support for API Governance," in *2014 IEEE International Conference on Cloud Engineering*, March 2014, pp. 615–618.

[21] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *J. of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2008. [Online]. Available: http://www.jmis-web.org/articles/765

[22] B. Hancock, K. Windridge, and E. Ockleford, "An Introduction to Qualitative Research," *The NIHR RDS EM / YH*, 2007.

[23] G. Guest, K. MacQueen, and E. Namey, "Applied Thematic Analysis," in *Introduction to Applied Thematic Analysis*, 2012, pp. 3–20. [Online]. Available: http://methods.sagepub.com/book/applied-thematic-analysis

[24] C. Auerbach and L. Silverstein, "Qualitative Data: An Introduction to Coding and Analysis," 2003.

[25] R. Ørngreen and K. Levinsen, "Workshops as a Research Methodology," *The Electronic Journal of eLearning*, vol. 15, no. 1, pp. 70–81, 2017. [Online]. Available: www.ejel.org

[26] J. P. Kotter, "Leading Change: Why Transformation Efforts Fail," *Harvard Business Review*, vol. 85, no. 1, p. 96, Jan. 2007.

[27] B. Burnes, "Kurt Lewin and the Planned Approach to Change: A Re-appraisal," *Journal of Management Studies*, vol. 41, no. 6, pp. 977–1002, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-6486.2004.00463.x

[28] A. Rüping, *Agile Documentation: A Pattern Guide to Producing Lightweight Documents for Software Projects*, ser. Wiley Software Patterns Series. Wiley, 2005. [Online]. Available: https://books.google.se/books?id=KdAEb0MOQCwC

[29] O. Rebollo, D. Mellado, and E. Fernandez-Medina, "ISGcloud: a Security Governance Framework for Cloud Computing," *The Computer Journal*, vol. 58, no. 10, pp. 2233–2254, Oct 2015.

[30] J. Cadle and D. Yeates, *Project Management for Information Systems*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2007.

[31] J. Karlsson and K. Ryan, "A Cost-Value Approach for Prioritizing Requirements," *IEEE Softw.*, vol. 14, no. 5, pp. 67–74, Sep. 1997. [Online]. Available: http://dx.doi.org/10.1109/52.605933

[32] J. Maxwell, "Understanding and Validity in Qualitative Research," *Harvard Educational Review*, vol. 62, no. 3, pp. 279–301, 1992. [Online]. Available: https://doi.org/10.17763/haer.62.3.8323320856251826

[33] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, p. 131, Dec 2008. [Online]. Available: https://doi.org/10.1007/s10664-008-9102-8

APPENDIX

A. First Iteration

1) Interview Questions First Round:

	Questions
1	What is your role in the company?
1.1	And your role in API the governance process?
2	Briefly describe your API governance process.
2.1	What are the benefits of it?
2.2	And the disadvantages?
2.3	What are the vital aspects or characteristics of that process?
3	Is it important to understand the effects of change to API? Why?
4	Who should be able to approve these changes?
5	How are requests handled? Would you handle them any other way?
6	Should changes be logged to enable traceability?
7	Is there a perceivable API ownership at your organization? Why?
7.1	If yes: Should these be responsible for managing it?
7.2	If no: Should a dedicated team be in-charge of it?
8	Is it important to have support from the organization's in the governance process?
9	Do you believe governance helps maintaining the integrity of an API? Why?
10	Does the governance process have an impact in the API life-cycle? How does it affect it?
11	We present the framework and prompt the interviewee for comments on aspects, strategies, team composition and roles involved

2) First iteration of the artifact:

Phases	
Assessment phase	current status quo. In this initial phase, the focus is on assessing the current status quo of API governance. From the framework overview, relevant aspects, strategies and key roles are identified. By identifying the points that the organization is lacking in, it is possible to move forward to the Implementation phase.
Implementation phase	the new status quo. This phase oversees the implementation of the aspects, strategies and participating roles provided in the overview. Personnel is appointed roles based on key skills. Aspects of governance are considered and analyzed, and applied to the governance process.
Termination phase	auditing the new status quo. This phase concludes the implementation by producing an audit on how API governance is now performed. Traceability is performed through documentation for future reference and other artifacts. These outcomes are available for re-use in later iterations. When the results are deemed as adequate to the organization's context, the iterative process should then be terminated. However, flexibility of the model allows for a re-evaluation of the status quo and re-start of the activities if necessary.

Aspects	**Supported by**
Change Control When changes are required, the produced effects shall be predictable and executed in a consistent method. Rollbacks to the change should restore functionality also in a consistent manner and complete manner.	[6], [20]
Impact of changes In the context of business, APIs are part of broad context and any changes to the API can have influence at a business level and IT operations. As such, stakeholder management, such as API consumers and business owners, shall be informed of these changes and what impact they create.	[6], [20]
Policy Specification and Analysis APIs as access points to assets shall only allow authorized clients access. API governance shall be concerned with access control policies, their analysis and application.	[6], [20]
Consistent Policy Implementation Governance over the policies that control the use of assets. APIs shall be implemented consistently independent of the technologies that is used to implement the assets themselves.	[6], [20]
Implementation Portability Governance over API integrity. As technology changes, API integrity must be preserved over implementations by decoupling it from the asset.	[6], [20]
Monitoring and Auditing API governance shall be concerned with monitoring and auditing API activity. The proliferation of APIs requires new approaches to control and govern APIs.	[6], [20]

Strategies	**Supported by**
Policies and risk relate to the correct way to do things. These strategies are also connected to risk control. Policy strategies ensure that there is compliance with laws, regulations, security control, risk mitigation strategies, corporate guidelines and industry best practices.	[17]
Operation Mode establishes a set of goals to guide the development of an API and enable the collection of metrics. For instance, one can refer to the business value an API provides; establish what conventions and standards to be used at development level.	[17]
Metrics are the means to provide visibility to governance. Usage, stability, and asset access downtime measurements can be contrasted with the determined policies, allowing for an understanding and evaluation of the governance program.	[17]
Processes provide the chance to develop and test an API and its compliance with the policies, as well as to make a decision on whether it should be accepted or not. These processes can be automated or human controlled.	[17]
People should be empowered to make decisions and oversee processes. Who can accept changes and who can request changes, and who can implement these changes are all an equally important.	[17]
Organization is responsible for nurturing a culture of support and reward for good governance over the APIs.	[17]

API development guidelines are good ways of ensuring that APIs are built up to a certain standard. Thoroughly documenting these guidelines ensures that APIs are built in a consistent manner, and can be worked on (developed, maintained) by different teams. Additionally, business related roles such as product owners can have a better view of the development process by consulting these documents.	[16], [17]
Catalog and classify different APIs different APIs, that way providing an accessible way of grouping APIs for instance with documentation.	[4], [16]

Governance Team:

Roles to be included	Supported by
• **Business line executives** • **API governance manager** • **Verification specialists** • **IT managers** • **Security managers** • **Auditors** • **Developer team** • **Testers**	[29]

Team Responsibilities	Inspired by
Governance principles What principles to follow to ensure the effectiveness governance. Regular assessment of compliance with policies and operation mode is conducted.	[8], [17]
Road-map and catalog Establish a road-map with a well defined time frame goals, and update it regularly, enabling an execution plan with milestones according to the road-map. When building the road-map stakeholder input should be added in order to identify goals that provide business value.	
Life-cycle model Create a model that captures the life-cycle of the API, for internal and external API users.	
Cross-role coordination Ensure that communication channels between the participating roles are kept open. Promote regular updates between groups.	
Responsibilities Clearly map roles to responsibilities so that everyone is fully aware of what they should really do, instead of what they believe they do.	
Decision power Organization should empower this team to make important decisions related to API governance.	
Ownership and accountability Map people to responsibilities. This enables a go-to-person to consult when understanding of some change or topic is necessary.	
Define metrics and KPI Quantifying business value and metrics increases progress and process visibility to the organization.	

3) Results:

Company One	
Role	The interviewee was a software architect involved in several and diverse areas.
Aspects	• **Simple** to introduce new platform versions into a project. • **Ownership** still leaning onto the maintainers. The governance board are mainly stakeholders who must approve changes to API. • **Integrity** of API is ensured by the governance process. • **Life-cycle** of API is independent of the governance process. However the governance process should be concerned about API life-cycle.
Strategies	• **Organization** support is essential for the governance process. • **Documenting** APIs helps to reduce the significant learning curve associated with an API. • **Transparency** from the governance process is a facilitator for developers, making it easily accessible for developers to understand the governance process and decisions in terms of API changes. • **Governance board** should focus on the entire system rather than components. Someone with expertise regarding abstraction and a broad vision of the system should integrate the governance board. • **Development** practices such as CI chain help analyze changes, their impact and outcome.
Challenges	• **Current work** dedication benefits will only bring be perceivable in the next version of an API. • **Documentation** requirements are hard to define. It is unclear whether new documents should be created with new requests or the existing documentation should be updated. • **Governance** process put extra tension in the project and developers.
Artifact Evaluation	The interviewee expressed that the framework aligns well their perspective of API governance. Most of the important areas were deemed as covered. The suggested aspects were in a more generic level, yet that was not seen as a drawback. It was also reinforced that API governance should be a non-ending process always looking for improvements. The main concern debated in this session was related to whether API governance should be detached from API implementation or whether these should be decoupled. The interviewee suggested that these should be decoupled.

Company Two	
Role	The interviewees were two software architects, who also have the role of interface guardians, that is, responsible for analyzing and approving change requests to interfaces.
Aspects	• **Backwards compatibility** should be ensured over a certain period of time. • **Ensure** work synchronization. That is prevent that people working in parallel do not end up modifying the same areas. • **Purpose** of API governance is mainly to prevent APIs to further "increase in size" more so than what is expected. Enforce the culture of taking the required extra steps to start a new API rather than constantly extend existing ones. • **Freedom** of change requests: anyone is allowed to put forward a request. However, it still has to be approved by interface guardians.
Strategies	• **Pre-study.** Performing a pre-study on change request allows for an early understanding of the impact and benefit to the system the change may bring. • **Versioning** of APIs as a method to catalog changes, related pre-study, reason that motivate the change, and its outcomes. • **Common agreement** of abstraction layer guardians and implementation layer guardians is required to ensure API integrity.
Challenges	• **Parallel work** has to be coordinated in the pre-study. Having a substantial amount of teams requesting and working on changes can easily create a bottleneck as they may end up attempting to work on the same areas of an API. • **Version control system** commonly used when parallel work is necessary is not ideal when a broad vision of all parallel activities is necessary. More documentation and clarity would be preferred. • **Balancing act** between resources and feature requests. This balancing act might suffer because of people' perspective. A project manager usually has limited resources so reuse might fit his objectives better, rather than spending the extra effort of starting something new. In the other hand an architect wants to protect the system and understands that if the system is expanded beyond what it is expected it is easy to lose control.
Artifact Evaluation	The interviewees had some reservations regarding the perceived hierarchy roles within the "Organization key participating roles" group present in the first iteration of our framework. The interviewees brought to our attention that an hierarchical context might vary depending on the organization's context, such us culture or country. The interviewees argued that a flat organization structure would probably make it difficult to map key participating roles extracted from an hierarchical point of view.

Company Three	
Role	The interviewee was a assistant engineer, with the additional role of approving change requests.
Aspects	• **Autonomy** for developers and architects to take decisions in terms of API changes without having to wait for approval from higher up roles. • **Hierarchical** decision tree from top to bottom. • **Audit and trail** via version control systems (such as Git) to manage and track changes. • **Integrity** is ensured by having someone to analyze compatibility resulting from changes.
Strategies	• **Empower** people to make change decisions but also encourage them to seek help in special cases. • **Inclusion** of architects in decisions. Usually, the architect role is involved in different areas of the project and can bring extra valuable knowledge. • **Foresee consequences** by doing a careful analyses of impacts changes may bring.
Challenges	• **Amount of requests** can become a bottleneck for the API governance board and developers. • **Size of organization** can affect the process. The same governance process might not fit all different sectors of the organization.
Artifact Evaluation	The concerns were directed to the proposed metrics as they should be adapted to measure the process itself rather than tracking API usage, as suggested in the first iteration of the artifact.

B. Second Iteration

1) Interview Questions Second Round:

	Questions
1	What is your role in the company?
2	Can you describe how an API governance process would ideally look like to you, in the context of your organization?
3	Who should be able to approve changes to an API?
4	There are claims in the literature that API governance should be detached from the actual implementation of the API. What are your thoughts on the subject?
4.1	What about decoupling API governance from the implementation of the asset the API interfaces with?
5	Present the Framework. (Stop after each slide, provide a brief explanation of its contents, and ask for the interviewee's feedback.)
5.1	Present the aspects.
5.2	Present the strategies.
5.3	Present the governance board concept
5.4	Present the governance board concept responsibilities

2) Second iteration of the artifact:

Phases: Updated the naming of one of the phases from *Termination* to *Reflection*.

Aspects	Supported by
Change Control When changes are required, the produced effects shall be predictable and executed in a consistent method. Rollbacks to the change should restore functionality also in a consistent manner and complete manner.	[6], [20] interviews
Impact of changes In the context of business, APIs are part of broad context and any changes to the API can have influence at a business level and IT operations. As such, the impact of changes shall be carefully evaluated. Stakeholders, such as API consumers and business owners, shall be informed of any changes and what impact they create.	[6], [20] interviews
Policy Specification and Analysis API governance shall be concerned with access control policies, their analysis and application. APIs as access points to assets shall only allow authorized clients access.	[6], [20]
Consistent Policy Implementation APIs shall be implemented consistently independent of the technologies that is used to implement the assets. Decoupling API from asset implementation allows for API integrity to be kept: changes to one do not influence the other.	[6], [20] interviews
Monitoring and Auditing API governance shall be concerned with monitoring and auditing API activity. The proliferation of APIs requires new approaches to control and govern APIs.	[6], [20] interviews
Life-cycle Alignment Life-cycle of an API shall not be decided by governance. However, the governance process is concerned with the API life-cycle. For instance, if the API is deprecated the governance process shall ensure that this API is not to be awaken again.	[6], [20] interviews
Safeguard API Integrity An API shall be able to interface on a newer version of the platform without conflicts, and without effort. When planning new features, existing API should not require extensive re-factoring, and backwards compatibility shall be ensured over a period of time.	[6], [20] interviews

Strategies	Supported by
Pre-study allows to foresee consequences of requests. A thorough analyses shall be conducted in order to understand the impacts of a request to the API. The outcome from the analyses is then used to assist the governance board with its final decision.	Interviews
Policies and risk relate to the correct way to do things. These strategies are also connected to risk control. Policy strategies ensure that there is compliance with laws, regulations, security control, risk mitigation strategies, corporate guidelines and industry best practices.	[17]
Operation Mode establishes a set of goals to guide the development of an API and enable the collection of metrics. For instance, one can refer to the business value an API provides; establish what conventions and standards to be used at development level.	[17]
Organization is responsible for nurturing a culture of support and reward for good governance over the APIs.	[17]
Metrics are the means to provide visibility to governance. The following are suggested metrics to consider: time for changes to be approved, number of approvals and refusals, stakeholder satisfaction scores, technical debt resulting from accepted changes, business impact resulting from accepted changes and service downtime caused by changes. These metrics can be contrasted with the determined policies and operation mode, allowing for an understanding and evaluation of the governance process.	[17] interviews
Acceptance Process provide the chance to develop and test an API and its compliance with the policies, as well as to make a decision on whether it should be accepted or not. These processes can be automated or human controlled.	[17]
People should be supported and encourage to make decisions and requests. Who can accept changes, who can request changes, and who can implement changes are all equally important aspects to consider.	[17] interviews
API development guidelines are good ways of ensuring that APIs are built up to a certain standard. Thoroughly documenting these guidelines ensures that APIs are built in a consistent manner, and can be worked on (developed, maintained) by different teams. Additionally, business related roles such as product owners can have a better view of the development process by consulting these documents.	[16], [17]
Catalog and classify different APIs different APIs, that way providing an accessible way of grouping APIs for instance with documentation.	[4], [16]

Governance Board:

Board Composition	Supported by
Those who understand the abstraction and those who understand the implementation.	Interviews

3) Results:

Company Four	
Role	The interviewee was a product owner.
Aspects	• **Simplicity** of dealing with internal requests. Different approach between small big changes. Product owner and developer team approve small changes, while big changes must be taken with product management. • **Internal requests** for small changes. Teams are allowed to request changes from other teams.
Strategies	• **Empower** the governance board to decide which changes shall pass through the governance process. Small changes can be handled by the development team and product owner by prioritizing task in the backlog internally.
Challenges	• **Difficult to test** all different combinations of modules that will integrate together, as it would require great effort. The product will be released as a big chunk, instead of partitioned releases. Governance over such a big product is something that the organization has yet to consider and investigate. • **A large scale API governance process** with many aspects and strategies to consider can be too heavy-weight for organizations without an in place process and that wish to start slowly.
Artifact Evaluation	Overall the interviewee expressed positivity towards the following aspects: Change Control, Impact of Change and Life-cycle Alignment. However, it was discussed that all other aspects seems necessary but too resource intensive to be applied all at once. Regarding strategies, positive feedback was received, yet the same issue was expressed regarding the heavy-weight nature of applying everything at once. For instance Metrics was deemed as valuable however it should be considered for a later iteration of the framework implementation process. Consequently, the concept of an iterative process was positively received as it allows for the expressed reservations to be tackled by implementing the aspects and strategies that align with the needs; "Yes I cant see any other way. We would probably fail if we take everything at once...". The interviewee further expressed positivity towards the API governance board, deeming it a good idea and well structured. Also denoted that it is important for the governance board to have representative roles from both abstraction and implementation. That way, ensuring that governance board is capable of a good understanding of the impact of changes.

Company Five	
Role	The interviewee was an engineer and system architect.
Aspects	• **API purpose** requires attentive consideration. Approval of changes to an API should be done by "...some kind of technical architect...". However and in order to preserve API purpose and integrity, there should also be someone responsible for the long-term vision, capable of discern and define what end goal an API must accomplish. • **API governance and API implementation overlap**, in particular for existing APIs. The people responsible for API development have the expertise, which in turn should be transparent to the organization to enable maintenance of the API over time.
Strategies	• **Guidelines for API technology** . Create and maintain guidelines for API technology and development should be a concern of the API governance process. Enable decisions to be made on what technology to use, ensuring API integrity and secure its business value.
Challenges	• **Business partners** integrating with the organizations products makes it difficult to carry out changes. Any changes must be carefully considered as to not harm the business with their partners. • **Introducing cloud solutions** extending the context in which APIs are currently used in the organization. • **Impact of changes** is a current struggle. Initially the focus was on how to implement an API, leading to issues such as loss of sight and lack of traceability on what the purpose of the API is. Additionally, there is a lack of understanding on how the business partners that use the API interpret the APIs purpose and how they use it. The result is having to "...go to our partners and ask them, how do you use this API?..". As a result, the organization must comply with how the business partners now interpret and use the API, which might differ substantially between partners.
Artifact Evaluation	Overall the interviewee expressed a high degree of approval, agreeing with all the presented aspects, strategies and API governance board responsibilities. However, one concern was expressed regarding strategies. The presented strategies are seen as posing a trade-off between the time expenditure versus the benefits in applying the strategies. "The problem is, making the compromise between these [strategies] and the product development, in your business where you want a short time to market...". The iterative nature of the suggested implementation process was deemed as having a natural feeling, similar to how the organization already works. Noteworthy, the interviewee asked for immediate access to the framework as there were parts they wanted to "...use right away...".

C. Third Iteration

1) Workshop Questions:

	Questions
1	What are the most relevant aspects?
1.1	What aspects would you ”pick and chose” to implement during the first iteration of the implementation process?
1.2	What advantages do you see?
1.3	And what disadvantages?
2.2	And the disadvantages?
1.3	Any additional thoughts?
2	What are the most relevant strategies?
2.1	What strategies would you ”pick and chose” to implement on the first iteration of the implementation process?
2.2	What advantages do you see?
2.3	And what disadvantages?
2.4	Any additional thoughts?
3	Regarding the Implementation Decision Model. Would you use such model?
3.1	Do you believe the model brings value to the framework?
4	Regarding the governance board. Any additional thoughts?
4.1	Do you agree with the member composition?
4.2	Should the governance board responsible for less/ more?
5	Regarding the Change Decision Model. Would you use such a model?
5.1	Do you believe the model brings value to the framework?
6	Would you consider applying the framework or parts of the framework?
6.1	Any final thoughts?

2) Third iteration of the artifact:

Aspects	Supported by
Change Control When changes are required, the resulting effects shall be carefully considered and executed in a consistent method. The long-term vision and purpose of an API should be identified and preserved. The pre-study strategy suggested by this framework helps to provide an overview of the expected outcome of a change. Rollbacks to the change should restore functionality also in a consistent and complete manner.	[6], [20] Interviews
Impact of changes In the context of business, APIs are part of a broad context and any changes to the API can have influence at a business level and IT operations. As such, the impact of changes shall be carefully evaluated. Stakeholders of an API, such as consumers and business owners, shall be informed of any changes and what impact they create.	[6], [20] Interviews
Policy Specification and Analysis API governance shall be concerned with access control policies, their analysis, and application. Who should access an API, and who should not is to be decided taking into account the business context. APIs as access points to assets shall only allow authorized clients access to the asset.	[6], [20] Interviews
Consistent Policy Implementation API governance should ensure that APIs are independent of the technologies that are used to implement the assets. Decoupling API from asset implementation allows for API integrity to be kept: changes to one do not influence the other.	[6], [20] Interviews
Monitoring API governance shall be concerned with monitoring API activity. The proliferation of APIs requires new approaches to control and govern APIs.	[6], [20] Interviews
Life-cycle Alignment The governance process is involved in all the duration of the API life-cycle. For instance, through monitoring API activity, if a decision is made to deprecate the API, the governance process shall ensure that it is not to be awoken again.	Interviews
Safeguard API Integrity An API shall be able to interface with a newer version of the platform without conflicts and without effort. When planning new features, existing API should not require extensive refactoring, and backward compatibility shall be ensured over a period of time. As an interviewee expressed, API governance ensures that “people do not do their own probably incompatible change between other changes”	Interviews

Strategies	Supported by
Pre-study allows foreseeing consequences of requests. A thorough analysis shall be conducted in order to understand the impacts of a request to the API. The outcome of the analysis is then used to assist the governance board with its final decision.	Interviews
Policies and risk relate to the correct method to work. These strategies are also connected to risk control. Policy strategies ensure that there is compliance with laws, regulations, security control, risk mitigation strategies, corporate guidelines and industry best practices.	[17] Interviews
Audit and trail for instance via version control systems enables tracking and management of changes to an API.	Interviews
Operation mode establishes a set of goals to guide the development of the API governance process and enable the collection of metrics. For instance, establish what scale of changes the API governance process should be concerned with small-scale changes versus large scale changes.	[17] Interviews
Acceptance processes provide the chance to develop and test an API and its compliance with the policies, as well as to make a decision on whether it should be accepted or not. These processes can be automated or human controlled.	Interviews
Metrics are the means to provide visibility to governance. The following are suggested metrics to consider: time for changes to be approved, number of approvals and refusals, stakeholder satisfaction scores, technical debt resulting from accepted changes, business impact resulting from accepted changes and service downtime caused by changes. These metrics can be contrasted with the determined policies and operation mode, allowing for an understanding and evaluation of the governance process.	[17] Interviews
People should be supported and encouraged to make decisions and requests regarding API changes and features. Who can accept changes, who can request changes, and who	[17] Interviews

can implement changes are all equally important aspects to consider.	
API development guidelines ensures that APIs are built up to a certain standard. Thoroughly documenting these guidelines ensures that APIs are built in a consistent manner, and can be worked on (developed, maintained) by different teams. Additionally, business-related roles such as product owners can have a better view of the development process by consulting these documents.	[16], [17] Interviews
Organization is responsible for nurturing a culture of support and reward for good governance over the APIs.	[17]
Catalog and classify different APIs different APIs, these procedures provide an accessible method of grouping APIs, for instance with documentation. API versioning could provide a unique identifier to unique states of an API, allowing for clear cataloging and classification.	[4], [16]

Governance Board:

Board Responsibilities	Inspired by
Governance principles Governance board should be concerned about what principles to follow to ensure the effectiveness of the governance process. Regular assessment of compliance with policies and operation mode strategy defined by the organization should be conducted.	[8], [17] •Interviews led to revision of definition
API technology guidelines Create and maintain guidelines for API technology and development. The governance board should be enabled to make suggestions and decisions on what technology to use in API development, securing API integrity and business value.	
Road-map and catalog Establish a road-map with a well-defined time-frame and goals, and update it regularly, enabling an execution plan with milestones according to the road-map. These could be goals related to pre-study completion, impact of change analysis or establishing an acceptance process for API changes. When building the road-map stakeholder input should be added in order to identify goals that provide business value.	
Capture Life-cycle model Create a model that captures the life-cycle of the API, for internal and external API users. If the API is deprecated, the governance board ensures that the API is not awoken again.	
Cross-role coordination Ensure that communication channels between the participating roles are kept open to all the interested stakeholders.	
Clear distribution of responsibility Clearly map roles to responsibilities so that everyone in the governance board is fully aware of what they should really do, instead of what they believe they do.	
Decision autonomy The organization should allow the governance board to make important decisions related to API governance. This approach helps to avoid potential bottlenecks by removing the need to report to a single person or "higher ups" from the organization.	
People traceability and accountability Associate people to their respective responsibilities. The concept of "who did what", enables a go-to-person to consult when the understanding of some change, or topic is necessary.	
Define metrics and KPI Quantifying business value and metrics increases progress and process visibility to the organization. These are tied to the Metrics strategy.	

3) Workshop Results:

Discussed Aspects
•Change control, Impact of Changes, Life-cycle Alignment, Consistent Policy Implementation, Monitoring, API Integrity.

Quotes
•"Without impact of change understanding nothing else matters".
•Monitoring is "the foundation to knowledge and change control allows one to be on top of development".

Discussed Strategies
•Acceptance Process, Pre-study, Audit and trail, People, API development Guidelines.

Quotes
•"These promote control and technical debt avoidance".
•"I see a lot of the strategies aims at control, and it is of course good to have control of what and why you develop something".

Framework Disadvantages
•Difficulty in reaching agreement over what changes to implement, what changes to govern and how to govern changes.
•Aspects are"people heavy when implemented".
•Lack of business strategies.
•Overhead.
•Heavy control may be a bottleneck, risking being so rigid that "nothing gets out".

Governance Board
•Well received and a positive addition.
•Having capable roles leads to "getting things to move forward".
•It can create a culture and environment for the product to evolve.

Decision Models
•Brings value to the framework.
•"Good to have".
•"We use a similar model today".

Would you apply this framework?
•One would most likely not, as the framework is "too focused on control" and they would prefer a more supporting role.
•Two out of three would partially apply.