Applying Goal Modeling to API Ecosystems: A Cross-Company Case Study

Bachelor of Science Thesis in Software Engineering and Management

Jamel Debbiche
Aksel Störmberg
Patrik Liao

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Applying Goal Modeling to API Ecosystems

Jamel Debbiche
Aksel Strömberg
Patrik Liao

Supervisors: Imed Hammouda, Jennifer Horkoff
Examiner: Grischa Liebel

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 2017

Applying Goal Modeling to API Ecosystems: A Cross-Company Case Study

Jamel Debbiche
University of Gothenburg
Gothenburg, Sweden
Email: jameldebbiche@gmail.com

Aksel Strömberg
University of Gothenburg
Gothenburg, Sweden
Email: aksel.stromberg@gmail.com

Patrik Liao
University of Gothenburg
Gothenburg, Sweden
Email: patrikliao@gmail.com

***Abstract*—APIs play a major role in software ecosystems and must continuously evolve to meet the demands of these ecosystems. In this paper we identify a new ecosystem around each API within software ecosystems and apply goal modeling to map such ecosystem. The authors collaborated with two software intense companies in a cross-case study. The outcome of this research was that by visualizing the relations between the actors within the API ecosystem, companies can better understand the ecosystem. An API ecosystem can be challenging to model and it is recommended that the mapping is done by an individual with experience in goal modeling. However, both, the modeling process and the mapped API ecosystem provide analytical benefits for an organization.**

***Keywords*-goal modeling; API; i* Framework; API Strategy, cross-case;**

I. Introduction

Application Programming Interfaces (APIs) have existed since computer programming first started, and experienced a boost in growth over the years [1]. An API is essentially an interface that allows an application to use services and functions of another application [2] hence facilitating the development process. More and more companies are considering developing their own APIs in order to offer access to their services and data (business assets) that will provide values to various stakeholders in their software ecosystems (SECO) [3]. This allows company to capitalize on the values that is shared in their ecosystem [4].

An API is considered as a necessary pillar that ensures the interaction between different actors in these SECOs [3]. A SECO can contain many APIs that cover many different needs through the entire ecosystem. However, in this work, we identify and focus on *API ecosystems*, defining these as ecosystems evolving around a single API within a SECO. Acquiring an API-focused perspective can allow organizations to continuously evolve the API that is needed to ensure the quality of the SECO as a whole. With this in mind, generating an API ecosystem model for every API can offer companies a strategic approach on a high-level perspective of the APIs.

A major difference between SECOs and API ecosystems is that SECOs do not explicitly focus on API stakeholders. Several modeling attempts towards SECOs map APIs simply as ”software resource” or ”platform” [5] which provide little depth on the functionalities, goals and quality attributes of the API. This limits the analytical ability necessary to develop an API and maintain it during the software ecosystem's life-cycle. An API ecosystem focuses on the API itself by modeling its direct stakeholders and considering its goals and quality attributes and as well as its dependencies with its actors. This detailed consideration of API stakeholders can increase the API quality and assist in its evolution along with the SECO it operates in.

More specifically, when analyzing the API value chain (see Fig. 1 [3]), one can see that a successful API must consider all relevant stakeholders and accomplish all the necessary goals that will offer values to the said stakeholders. This high-level perspective of an API shows a complex structure surrounding the interface. This complex construct contains the relevant stakeholders as well as the services and data exchanged and how it is shared amongst the actors. Actors can be stakeholders, software or hardware components that are set to achieve a certain goal(s) that will ultimately increase the quality of a SECO [3]. There is currently no research on software ecosystems focusing on APIs, but given the rise of SECOs and APIs in the software industry, it is important to recognize and explore these API ecosystems.

Mostly, research on APIs focuses on low-level designs, such as API development best practices that are usually language specific [6]. There is little research on APIs that addresses high-level design in a software ecosystem context. Previous research [3] shows that APIs are not merely a technical tasks and requires architectural and design decisions in order to continuously empower the broader software ecosystems.

Software ecosystems are considered a complex structure that are challenging to create and maintain [7], in order to facilitate the understanding of such ecosystems, mapping strategies have been introduced and several attempts have been made in modeling SECOs to provide different advantages [8], [9]. For instance, i*, a goal modeling language, was used to present actors, goals and dependencies between these elements. This methodology was helpful in providing structure to a complex system and this mapping assisted in providing insights and taking strategic decisions [8] when it comes to developing their ecosystem further.

Similarly to modeling SECO [9], mapping an API ecosystem can provide an overview of how an API ecosystem looks. This can provide advantages when it comes to strategic reasoning that can allow companies to make strategic decisions that can improve their API ecosystem. This thesis addresses the gap in research when it comes to API ecosystems and API ecosystem mapping and evaluates a predefined modeling language to determine if it can assist companies that are faced with the need to analyze and form such ecosystems.

Looking at the modeling languages used to map SECOs, some of them like, the Software Ecosystem Modeling (SEM) technique, gives a holistic view of the object in the ecosystem [10] but does not focus on different actors and their relationships, which is an essential part in an API value chain. Goal modeling, however, allows the investigation and analysis of actors, their dependencies and goals that gives the companies a better understanding of the possible strategic decisions and alternative configuration of a SECO [8]. This is done by mapping actors and their relationships along with the relevant resources and tasks taken by each actor [8].

Therefore, goal modeling caters well to all the components of an API value chain. This modeling technique allows the mapping of all API actors, such as API users, consumers and providers. It also allows for expressing the goals these stakeholders when it comes to collaborating with different actors [9]. These benefits are all expressed in Yu's application of this modeling tool to software ecosystems [8].

In order to examine the usefulness of mapping an API ecosystem using goal models, we will model the upcoming API ecosystems of two case companies. The modeling will be an iterative process where the companies will continuously provide feedback of on the API ecosystem model created. This research is conducted as part of a Software Center project.

Providing a methodology and identifying best practices of API ecosystem mapping can improve the analytical and descriptive effectiveness of the modeling techniques [11]. The findings of this thesis will also contribute to the academic community by being first to introduce the API ecosystem and apply goal modeling to map such ecosystem in an industrial context. This may also be helpful for modeling SECOs, as there are currently few studies on modeling techniques and methodology [11].

Furthermore, examining strengths and weakness of mapping API ecosystem can help identify the analytical capabilities and limitation of goal modeling in an industry application that is otherwise lacking in the academic community [11]. This knowledge can also assist companies in deciding whether or not goal modeling will facilitate their decision making in terms of API ecosystems.

A. Purpose

The purpose of this research is to assess the usefulness of mapping an API ecosystem using i* as an example goal modeling language, the choice of this framework will be motivated in the literature review. In addition, we will explore best practices when it comes to creating API ecosystem mappings, in general, and specifically with goal models.

1) Research Questions: The main research questions are:

- RQ1: How can goal models be applied to model API ecosystems?
- RQ2: How useful is mapping API ecosystems using goal model?
  - RQ2.1: What are the benefits of goal modeling API ecosystems?
  - RQ2.2: What are the drawbacks of goal modeling API ecosystems?

II. Literature Review

A. Modeling Software Ecosystems

A software ecosystem entails interconnected software platforms that operates in alignment with a company's goals, along with its relationship to different stakeholders (actors). For a software ecosystem to be considered successful, it has to bring value to all its stakeholders [3].

Modeling an ecosystem helps identify and analyze complex relationships [8] that are otherwise hidden or missing from other documentations such use cases and user stories. Correctly identifying the important components in a software ecosystem is crucial since all stakeholders success is directly related to the overall quality of the ecosystem [3]. Without clearly understanding all the dependencies and connection, designing an ecosystem becomes extremely challenging [8].

When it comes to modeling ecosystems, several tools have been used for mapping SECOs, with each tool examining a different perspective. For example, the Software Ecosystem Modeling (SEM) technique which includes the Product Deployment Context (PDC) modeling language that was formed to give a simple holistic overview that is limited to the dependencies between software products within the ecosystem [12]. The Software Supply Network (SSN) that maps the software, hardware and services developed by companies that are attempting to satisfy market needs [12]. While SNN seems to provide an overview between the dependencies between organizations, it only considers the actors that are supplying the products and dismisses other relevant stakeholders such as the customer that can be a crucial actor to a software ecosystem.

Furthermore, Business Model Canvas (BMC), a textual representation of high-level business view has been used to model a software ecosystem focusing on who and how the business value is generated [13]. However, this representation does not give any value to software developers and does not provide any advantage to software developers, nor does it provide any analytic benefit for the organization [11]. e^3Value Modeling technique also shares the same limitation by targeting primarily the business perspective of an ecosystem, however, unlike BMC, e^3V does provide a visual representation of the SECOs [11].

In addition, Value Network Diagram (VN) has also been used to model SECOs, this language models the value exchanges between human actors which also makes it a modeling language heavily focused on business aspects of these ecosystems [14]. Hence making it unsuitable for software developers. Additionally, VN also lacks formal syntax as well as tools used to develop such models [11].

In this paper, we relate to the actors mentioned in the API value chain, meaning that the business assets, APIs, End-users, developers and applications using the API must be modeled in order to show a correct representation of an API.

B. APIs from a High-Level Perspective

Application programming interfaces are has been mostly seen as technical resources that allow third party actors uses data and services of an organization. However, a recent examination of API design from a SECO perspective have negated this idea [3]. In an ecosystem context the API must supply the API user with the necessary functionality to deliver business values to the end customers. It must also be agile enough to support the rapid changes that are occurring in its ecosystem. Therefore, an API is considered an important aspect of a company's software ecosystem as it connects to many actors; for instance, business assets as well as several stakeholders. In other words, all actors in a software ecosystem can be connected and share companies' services through APIs [3].

Additionally, companies can use APIs in their software ecosystem to attract new actors to their platform and increase brand awareness which in turn will generate business value to all involved stakeholders. This occurs when actors use an API for the first time and decide to make it a bigger part of their development platform and hence maybe become a more established actor [3]. This is illustrated in Fig. 1, which shows the ecosystem aspects around an API to a model introduced by Jacobson et al. [15].

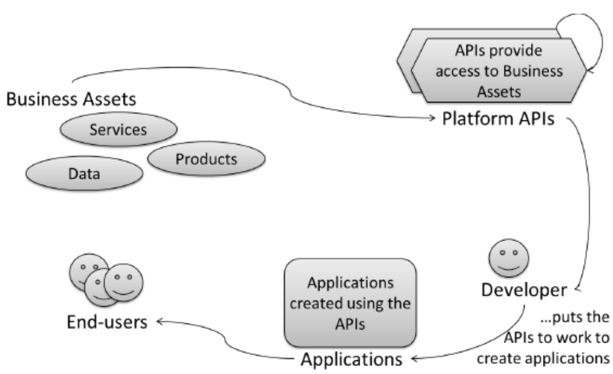

Fig. 1. API value chain in the context of a SECO [3]

A recent study applied i* strategic modeling to software ecosystems and concluded that the visualization helped make relationships more explicit for the actors of the ecosystem. Moreover, the modeling helped exploring strategies and alternative decisions by structuring the complex environment [8]. When it comes to APIs, this modeling can be helpful at visualizing all the important elements (actors, dependencies etc.). Additionally, this can allow companies to be clearer in understanding the value chain of an API and therefore, be able to get an in-depth analysis of the ecosystem [16] and take more strategic decisions.

When mapping an API ecosystem, it is important to consider the value chain of an API in order to satisfy all actors in the ecosystem. A previous research within industry-academia collaboration [17] identified the API as one layer within a larger multi-layered construct as seen in Fig. 2 derived from the API value chain [3]. The construct contains four layers; Business Assets, API, Application Software and Domain [17]. And between every layer, there is an element that impact its neighboring layers, for instance a use case will affect both the product and how the API is used. Below, we provide a brief explanation of every layer:

- Domain Layer: This layer concerns certain needs that support the Application Software and delivers a value to the end-user. Here the previous study [17] recommends the examination of use cases and business models to fulfill this layer.
- Application Software Layer: This layer is where the features of an API become visible. To fulfill this layer, it is important to consider whom are the consumers of the API and what features are used in the application.
- API Layer: Here we identify more lower level aspect of the API itself, for instance; the type of responses provided by the API (synchronous or not) and how it should handle unauthorized access.
- Business Assets Layer: Company's assets such as data are of concern here. This layer should state which assets are exposed and to whom and how they are exposed.

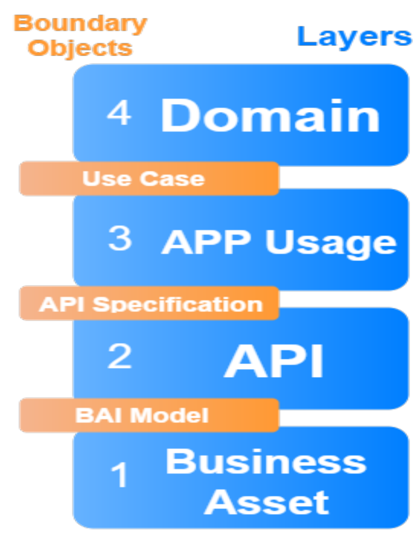

Fig. 2. An API in relation to relevant layers [17]

As seen in Fig. 1, the API value chain is mostly concerned with which actors to include; who provides the API, who consumes the API, and who will use the end-product. Additionally, these layers, as well as the value chain, stressed on the needs requested from these actors and how to satisfy their needs. Different actors have different demands that needs to do be addressed differently, and it is up to the company to take the design decisions necessary to fulfill them [3].

C. Goal Modeling & the i* Framework

1) Usage of the i Framework:* Goal modeling focuses on mapping goals of actors in a software context, usually early requirement engineering [18]. Apart from mapping the relevant stakeholders and their goals, it also models non-functional requirements as well as tasks. In addition goal modeling also displays the interconnection between these elements [19]–[21].

The i* framework is one of many languages used for goal modeling. i* uses graphical representation to describe the dependencies between different actors, this is known as Strategic Dependency Diagram [22]. i* also targets the intentions of every actors and how they can be addressed either internally or via dependencies to other actors. This is knows as Strategic Rationale Diagram [22].

The syntax of i* framework is summarized in the Fig. 3 :

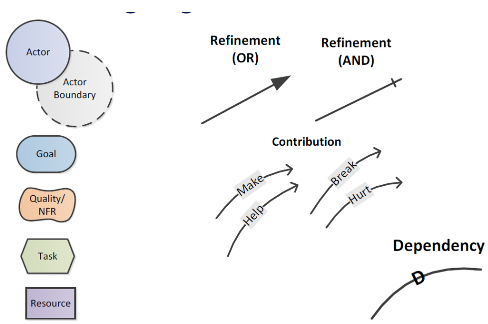

Fig. 3. Goal Modeling Syntax [23]

From Fig. 3, we explain the following i* goal modeling components [23]:

- Actors: Actors are entities that work to achieve certain goals, often in cooperation with other actors.
- Actor Boundary: Actor boundaries show which actors' goals, qualities, tasks and resources belong to an actor.
- Goal: A specific, well-defined objective that an actor want to accomplish.
- Quality (or soft-goal): A quality attribute that an actor wants to achieve to a certain level. For instance, security may be a quality that an actor wants to increase.
- Task: An action that an actor takes to serve a certain goal or quality.
- Resource: An asset that an actor uses to perform a certain task.

All of these components can be connected with the following links:

- AND link: The AND link shows that the completion of all the children components is required for completion of the parent component.
- OR link: The completion of any child component will fulfill the parent component
- Contribution links: These links shows the degree of effect a certain elements have on a quality component. For instance, adding facial recognition feature Helps the Security quality.
- Dependency link: This link shows how a component of an actor is dependent on a component of another actor. This link is has five arguments:
  - Depender: The actor that depends for something for something to be provided.
  - DependerElmt: The element within the depender actor that indicates where the dependency starts from.
  - Dependum: The elements that the depender depends on.
  - Dependee: The actor that the depender depends on.
  - DependeeElmt: The element inside the dependee that provides / accomplish the dependum.

To better understand goal modeling, we present a simplified example in Fig 4. In this scenario, we are mapping a Customer that uses a Camera App to get the number of people that are in a building. In the example, we model two actors; Customer and Camera Client App. The Customer actors has one goal of "Knowing number of people in Building", the actors also have two soft-goals (or quality attributes) in which he/she wants to satisfy. First being able to know the number of people in the building accurately and easily access this information. To accomplish the goal, the actor has two possibilities:

- Task 1: User Camera A Client App, this means that the customer will rely on the Camera actor (more specifically, its function to calculate numbers), in order to know the number of people, hence the dependency link between the two. In this example, the customer will depend on the getting the resource "Num individuals in the building in building". This is modeled in between the actors as a dependum.
- Task 2: The other solution to accomplish the goal is to use alternative (and not depend on the Camera Client App). This is represented as a task "Use other alternative eg., turnstiles". In a more complete model, this task will depend on a Turnstile actor.

If the actor chooses the Task 1 to accomplish the goal, then the camera app will enable him/her to easily access the data, hence the Make link. However, the camera may not be as accurate given its limited field of view for example. The customer can prioritize accuracy and use turnstiles however, a turnstile does not offer easy access to data, hence the Break link between the two elements. Here we can the alternative that the actor has, along with the trade-offs of each option.

Similarly to the Customer actor, the Camera actor is has a goal to count people from the camera, this is accomplished by two tasks "Get Data from Camera" AND "Calculate Number".

2) Advantages and Limitation of the i Framework:* On the contrary to some of the modeling language mentioned above, the i* framework are intended to provide analytical advantages to organizations by visualizing the dependencies between the actors [24].

Additionally, unlike the previously mentioned modeling languages, goal modeling supports refinement and traceability, meaning the model can contain different level of details and information. This can be useful when it comes to examining

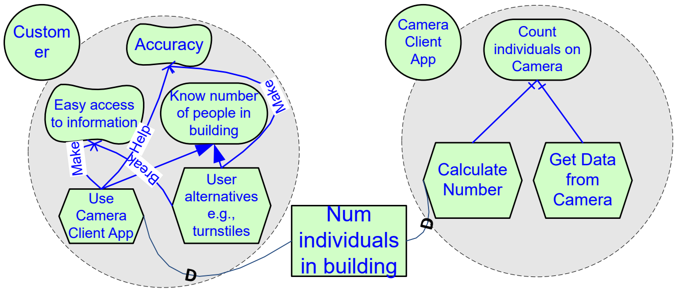

Fig. 4. Example i* Model for a Camera Client App

the ecosystem from different perspectives [11]. Moreover, this technique do have a formal syntax unlike SNN, VN and e[3]V.

Nonetheless, the i* framework relies solely on visual representations, this means that its users must be able to understand the syntax to take advantages of its usefulness. In addition, like most of modeling languages, the i* framework lacks applications in the industry context and there is no established methodology on how to apply it [11].

III. Case Companies' Description

This research is a cross-case investigations of modeling an API ecosystem in an industrial context. Primary data is collected, analyzed and compared in order to answer the given research questions. The two companies chosen are software-intensive, and the data collected serves as the foundation of this study. These two companies were selected as they are involved in a Software Center project. In this section, the two companies are briefly introduced.

A. Tetra Pak

Tetra Pak is a company that originated in Sweden and provides primarily packaging for liquid and food products but also a range of processing and packaging technologies in a broader array of products. Additionally, the company supplies databases that contain several key information in which their customers use to generate reports about their manufacturing tasks such as quality control reports. Tetra Pak can also be the authors of these reports as well. In order to make the generation of these reports easier, the organization aims to provide an API where the authors of the said report can use to access the needed data easily and generate the report much faster. The API will help the customers and the company create and generate reports more efficiently. According to Tetra Pak, over 70% of their customers are unable to generate their reports themselves[1]. In other words, Tetra Pak needs to take over and develop the reports demanded by their clients.

The company provided the following people as participants in this research:

- One Development Engineer
- One Senior SW Development Manager
- One SW Engineering and Product Manager

[1]According to a participant during the Introductory Group Interview

B. Axis Communications

Axis Communications is a Swedish-based company and the market leader in network video. They are the inventor of the first network camera in the world and is currently providing network video products which are installed in public areas such as train stations and universities, as well as business areas such as casinos and retail stores. Axis Communications is attempting to add value to their cameras by providing an API to their customers to enable them to create their own applications where they use the data generated from Axis' network cameras. The company aims to provide their customers with the opportunity to customize the functions of their Axis network cameras, and believes that the a Cloud API will serve the needs of these customers and thus increase customer satisfaction and customer retention.

The company provided the following people as participants in this research:

- One Expert Engineer
- One Technical Product Manager

IV. Methodology

This research aims to understand API ecosystems and develop a method for modeling these ecosystem using an existing modeling tool, in this case, goal modeling. In order to explore this phenomena, this research takes a qualitative approach [25] by interviewing two case companies that currently aim to create their APIs. Adopting a qualitative approach, allow us to understand the interaction and intentions of all important actors and how different components depend on each other inside an API ecosystem.

A. Data Collection

Given the newness of API ecosystems, and as this study aims to investigate this construct in an industry context, interviewing a group of employees building an API can be a very beneficial way to collect data. This is because group interviews facilitate discussions about the topic studied and these discussions are usually directed using open-ended questions [26] that are commonly extracted from data gathered before the interviews. These open-ended questions will give the participants a chance in engaging in discussions that interest them and hence generate a more organic and valuable data.

To ensure the collection of rich data, these interviews must take place in a controlled environment where the participants feel comfortable and safe to initiate and maintain discussions [27]. The interviews were conducted in the company's' facilities, and the names of the participants remained confidential, this increases the comfort of the participant. Moreover, in every group interview, the number of participants were between four and eight, which is suggested by Kitzinger as larger groups are more difficult to manage and smaller groups may offer shorter discussions and some data may be left unexplored.

We conducted two group interviews and one interactive workshop with each company where all of the interviews were recorded with their consent:

1) **Introductory Group Interview**: This interview was necessary to understand the API ecosystem as a whole of the companies.
2) **Interactive Workshop**: This was necessary to gather data to model the ecosystem of the companies and also to explore the correct set of modeling an API ecosystem. This assisted us to answer RQ.1
3) **Extra Online Interview**: An additional interview with Tetra Pak was needed to finalize data collection needed for RQ.1 as further gaps were identified after the workshop. (only a development engineer was present).
4) **Online Group Interview**: This interview was conducted to discuss the benefits of modeling an API ecosystem (RQ2).

1) Introductory Group Interview: At first, a group interview was conducted with each company in order to introduce the subject of API ecosystems. More specifically, the companies were familiarized with the API value chain in order to have a more holistic view of an API ecosystem. We also introduced the API as a boundary object surrounded by four relevant layers as explained above. The companies had participated in a previous Software Center Sprint, so they were already a bit familiar with ideas concerning strategic API design.

This initial group meeting also allowed both companies to explain, in a general sense, the aim of their APIs and how it will work in relation to other actors in the ecosystem. The data collected from this interview was not conclusive enough to start modeling given the generality of their explanation, however, it provided us an overall understanding of the companies and their API needs, and this was useful for planning purposes. Additionally, understanding the context of the companies' API ecosystem is crucial to the modeling process.

2) Interactive Workshops: After the first introductory group meeting, we conducted one interactive workshop with each company. They were introduced to the basic syntax of goal modeling at the beginning of the workshop, this is done by presenting an example of already-modeled use-case and explaining the roles of every element and link used in the model.

Prior to the interactive workshop, when attempting to model the documents provided by the companies (use-case and user-stories), we discovered many missing dependencies between goal modeling elements. In other words, the paper provided did not provide all the necessary data to model the API ecosystem. These gaps produced open-ended questions and were used as a guide in the interactive workshop, a sample of the list of questions can be found in the Appendix under "Sample of Tetra Pak Interactive Workshop Questions". In addition, comments from the workshop can also be found in the Appendix under "Interactive Workshop Data".

The interactiveness comes as the participants cooperated in modeling the API ecosystem. The questions were asked to spark a discussion about the gaps identified during the initial modeling, and as the participants explain these gaps and uncertainties, we model the API ecosystem accordingly in front of them as they continue to provide feedback on the modeling done. The participants mentioned in section IV were present on every workshop while other staff members that are also concerned with the API ecosystem joined in occasionally. These interviews, along with the modeling process allowed us to develop a method of mapping an API ecosystem derived from modeling both companies ecosystems, hence answering RQ1.

3) Online Group Interview: Information gathered from the introductory interviews and the interactive workshop was needed to understand and accurately model the companies' API ecosystems. This data was useful to learn how to apply goal modeling to API ecosystem, hence answering the research questions. However, to get further feedback on the models and to answer the second research questions we conducted an online group meeting with each company in order to discuss the usefulness of the model created. In other words, what benefits and drawbacks do the models present for the companies. The questions asks were open ended and some questions were extracted from the modeling process, of instance *"To what extend was using color red helpful to highlight the most elements causing problems?"*. The full Interview Guide can be found in the Appendix under Online Interview Guide. Data collected from these interviews are used to conclude benefits and drawbacks of the API ecosystem modeling, hence answering the second research question.

4) Modeling Methodology: After the first interview, the case companies were asked to share user-stories and use-cases as a starting point to allow us to begin modeling their API ecosystem. These user-stories and use-cases were analyzed and goal modeling components were extracted from the documents, was illustrated in initial drafts. Tetra Pak offered two user-stories that they plan to implement in their API, and Axis Communications shared a single use-case as a starting point.

Collecting data was iterative, there has been a constant contact with companies and the supervisor in order to make sure the modeling was done in a syntactically-correct manner and also to make sure the models were correctly mapping the companies' API ecosystems. This constant contact meant that the models were constantly revised and several versions were produced that will be presented below.

After the interactive workshop, we applied all the comments given by the companies and sent the updated versions of the model to them by email in order to make sure the API ecosystem was properly and completely mapped. As explained above, we maintained constant contact with the supervisor and the companies to ensure constant flow of feedback that enabled us to continuously improve the models. This iterative process is illustrated in Fig. 5. All of the modeling was done in Microsoft Visio using the i* framework.

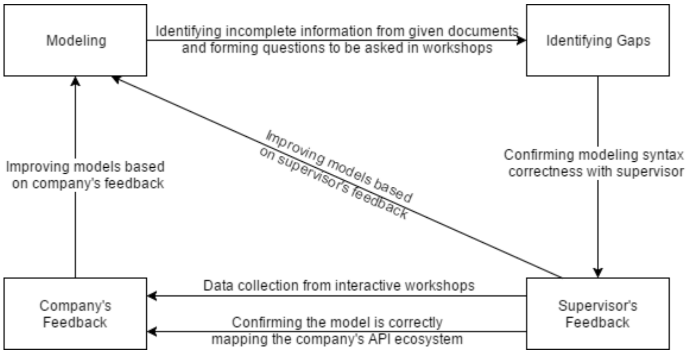

Fig. 5. Iterative Modeling Process

B. Data Analysis

While all the interviews and workshop were recorded, only the online group interviews were coded. This is because the other group interviews were be used to develop and update the models created, hence the data gathered from them were directly used as feedback in the modeling process. The online group meeting however, were used to answer the second research question: the benefits and drawbacks of modeling API ecosystems.

While qualitative data can be analyzed in many different ways, most approaches , if not all, includes coding and identifying patterns [28].

As seen in the interview guide in the Appendix under Online Interview Guide, the questions targets four main areas:

- Applying Layers to the API Ecosystem;
- Highlighting Area of Interests;
- Complexity of API ecosystem models (learning curve);
- Alternative design languages.

These aspects were directly derived from our modeling process, denoting the most important iteration of creating the models. The interview data was transcribed and categorized under the areas listed above and several variable were extracted from this process:

- Usefulness of layering the API ecosystem models;
- Usefulness of applying colors to area of interest;
- Complexity of the models;
- Complexity of the modeling language.

All of these variables were coded into five levels: very low, low, medium, high and very high. All segments of the transcripts that contribute in describing the level of these variables are listed in the Fig. 13 under the Result section. Please note that the table present a small sample of the entire interview transcript. For the complete transcript, please refer to the Appendix.

V. Validity Threats

A. Construct Validity

One threat to construct validity can be that the participants do not interpret the questions and terms such as API ecosystems, goal modeling and relevant API ecosystem the correct way. This threat is reduced by introducing these concepts at the start of each interview, and also including a description of the layers when sending different versions of the models to the companies.

Additionally, the questions used during the group interviews were shared with the supervisor in advance to ensure their correctness.

B. Internal Validity

When it comes to internal validity, we were able to conduct group interviews of equal lengths to collect same amount of data from both companies. Additionally, the questions were gathered in the same manner (from the documents provided by the organization). Moreover, we used the same material to introduce goal modeling and relevant API layers to companies to make sure the same amount of knowledge is shared with all the participant.

However, one threat to internal validity is that we had an additional online group interview with Tetra Pak for further modeling, this was not needed with Axis, however, spending more time modeling maybe have produced slightly different results. Spending more time with the modeling process can help identify more gaps as the all the iteration is shared and discussed with the supervisor.

Additionally, the documents shared by the two organization differed in length, and modeling Tetra Pak's API ecosystem started one month earlier.

C. External Validity

Much like SECOs, API ecosystems are context-dependent, meaning companies develop specific APIs to serve a specific need, hence the ecosystem will differ from a company to another. Therefore, modeling the API ecosystem of a third company may differ, especially if we were given different documents in the beginning of the process. This research takes a first step in modeling API ecosystems, in order to obtain more generalizable results, the modeling methodology must be applied to more companies.

D. Reliability

In order to reduce the threats to reliability, we have been in constant contact with the supervisor and the companies to ensure that the steps taken are comprehensive and properly documented. All the interview guides, except the Online Group Interview Guide, received feedback and were improved accordingly.

VI. Results

A. Applying Goal Modeling to API Ecosystems

This section explains the steps taken when it comes to modeling the companies API ecosystems. While reading the documentation given by the companies, we identified and discussed every actor mentioned along with its tasks and goals, as well the resources. The modeling process started once the documents were received till the last interview with the companies. The documents received and the data collected during interactive workshop were both used to answer the first research question.

Please note that in this section, all the figures presented has been re-purposed to show a high-level overview or certain areas of the models. Additionally, some models may be unreadable in this section due to its size. Therefore, for the complete models, please refer to the Appendix.

1) Modeling Tetra Pak's API Ecosystem: Tetra Pak shared two user-stories in the beginning of the project, this gave us approximately one month to create and refine models before the interactive workshop.

The user-stories entails information about different types of reporting done in manufacturing plants, more specifically, one user-story involves Quality Control (QC) reporting and the other describes Clean-In-Place (CIP) reporting. The API will be used to facilitate the report creation and generation. Currently, finding data in Tetra Pak's database is a challenging tasks, therefore, these customers will often rely on Tetra Pak Market Companies to create customer reports. Additionally, these Market Companies also rely on the central part of Tetra Pak, responsible for the development of the API (called Tetra Pak Development in the model) on assisting them in finding data.

Each user-story was developed on a separate model to make sure nothing important was left out from the user stories' documents. Once the models were complete, we confirmed their correctness with the supervisor and Tetra Pak during the interactive workshop. The first model produced, based on the first user-story, was developed by the supervisor in order to illustrate the correct way to use the i* framework. We later modeled the second user-story. This resulted in the creation of the models in Fig. 6 and Fig. 18. As explained above, the modeling was an iterative process, which means several model version were produced for each step, all the versions will be included in the Appendix while we will display the most important iterations in this section.

In Fig 6, we modeled the first use case where seven actors were identified: Customer, CIP Execution Program, Operator, TPM Product Integrator, Actor, Tetra Pak, Business Intelligence Tool and Unknown?. These actors were directly extracted from the CIP user-story. The Unknown? actor is the result of a gap identified while examining the user-story document, as the company does not specify who exactly will provide the Expert Knowledge resource to the Tetra Pak actor.

In the figure, the customer (the factory) has a goal of producing CIP reports (represented as goal) which can be done from a user interface. The customer can also acquire these CIP reports using PLCs without a common structure (an alternative method using a different method of generating CIP data). This task depends on the Operator, hence the dependency between the two actors. Moreover, the Operator can also be considered a customer, hence the ISA (Is-an-association) link, meaning it is a specialized case of the Customer actor [23]. We also observe several soft-goals in the Customer actor and Tetra Pak actor. For example, the customer aims to save costs and reduce energy by acquiring the said reports. Tetra Pak also aims to collect data from the customer (represented as a task) in order to analyze and optimize the customers CIP performance. Hence, the customer is dependent on Tetra Pak to increase performance and decrease costs.

Once the user-stories were correctly modeled to our understanding of the documents, we explored different way to map these elements in relations to API layers explained in the Literature Review [3]. This was firstly done by visualizing four layers in the model and attempting to place every actor in its appropriate layer. For instance, the database actor will be placed in the Business Assets layer as it serve as an asset to be accessed, and so forth.

Layering the first use-case was possible as seen in Fig. 17 in the Appendix, however some actors from the second user-story could not be properly placed in a layer, this is because these actors may play roles in more than one layer depending on the timeline of the API ecosystem. For instance, in the near future, the customer will not use the API, however, the company aim to open the API to its customers in the distant future, therefore, the customer will be both. part of the domain layer and the APP SW layer. In this case, as seen in Fig. 7. Hence we needed to find another way to represent layers in the API ecosystem.

The second attempt at placing actors into API layers was attempted by color-coding the layers, meaning each layer was assigned a color, and every component was assigned the color(s) of the layer it represents. As the previous attempt, the limitation in this technique was that many components were representing more than one layer. This means that the components will be assigned two colors, which was not possible to represent in Microsoft Visio as elements cannot have more than one color.

Attempting to layer the model generated more questions regarding the API ecosystem, and from the model we were able to observe additional missing links and contextual inconsistencies. These questions regarding gaps helped spark discussion within the company in the interactive workshop, especially about the future of the API.

Before proceeding with improving the model, we needed to fuse the two user-stories in one model that is representative of the API ecosystem as a whole and not specific to a single user-story. In other words, we generalized every user-story-specific component in order to extract a single model that represent the entire API ecosystem. This was after the interactive workshop is shown in Fig. 20 in the Appendix.

During the interactive workshop, we were able to receive more data regarding the API ecosystem by discussing the identified gaps. However, given the time required to model, not all feedback were added during the workshop. Therefore, after the workshop, the modeling continued and more gaps were generated, as seen in Fig. 8.

After the workshop, we adjusted the model in relation to the feedback received and we made another attempt to represent the API layers by mapping every layer in a separate model.

During the layering process, we learned that some actors did not belong to any relevant API layer, and as a result, we

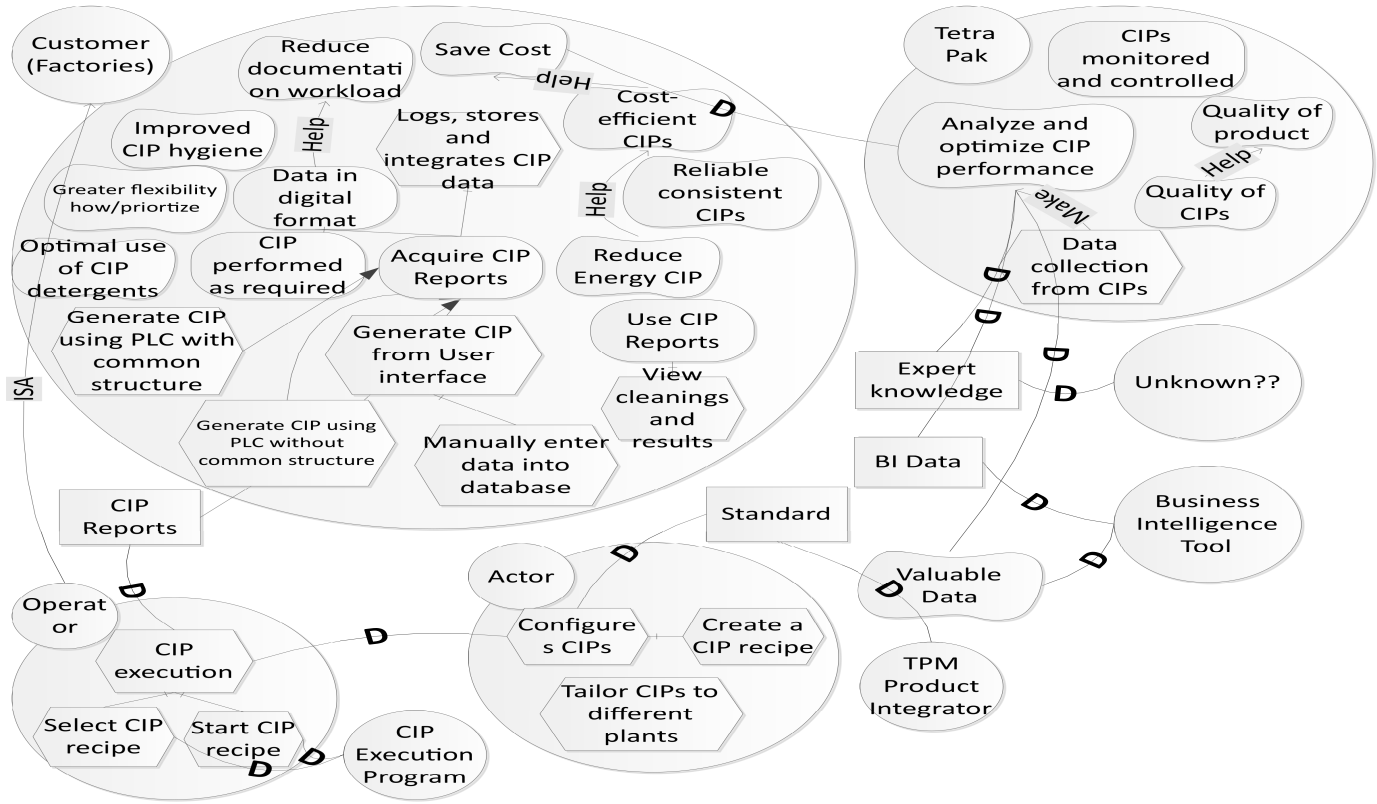

Fig. 6. Model version 1 of the first user-story

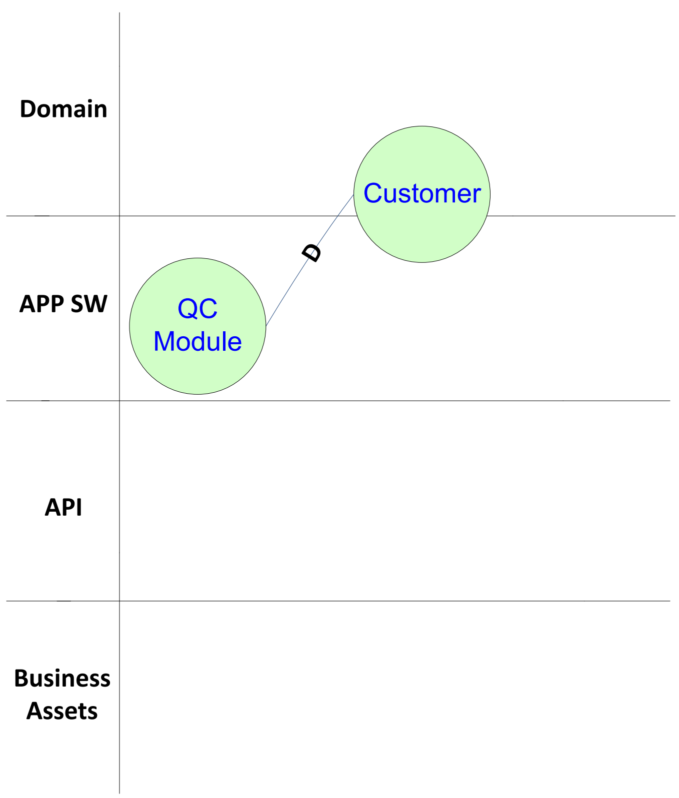

Fig. 7. Attempt 1 in Visualizing Layers in the second user-story

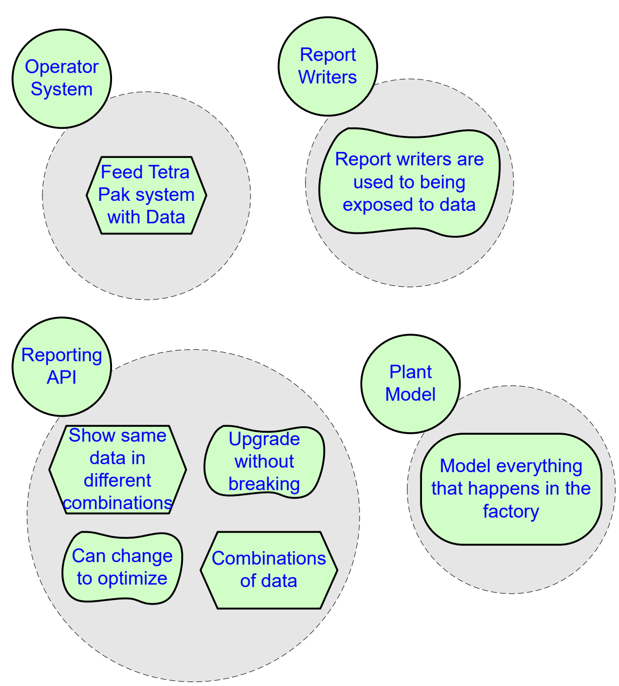

Fig. 8. Cropped Section of Incomplete Tetra-Pak API Ecosystem

needed to create two new layers that did not directly relate to the API layers: ”Future Layer” and ”Pre-report Generation layer”. The former displays the actors whose activities will change over time, and the latter maps different ways Tetra Pak collects data from the customer prior to the report generation.

Having a separate model for each layer allowed us to examine each layer independently others. For instance, this made us question the relevance of every layer and every component to the overall API ecosystem. An example of this, is the ”Pre-Generation Report” and the ”Future” Layer showed no relevance to API ecosystem and hence was removed, making the model less cluttered and more concise. The four layers: Domain, APP SW, API and Business Assets are displayed in the Appendix Fig. 22, 23, 24 and 25 respectively.

Having the layers in separate documents refrained us from analyzing how different layers depend on each other. The limitation of this view forced us to return to visualizing layers in one model, in other words, the entire API ecosystem must be mapped in a single model. The challenge, in Tetra Pak's case, was that actors tasks and goals changes with time hence actors will belong to different layers. Which led to further discussion with the company.

We proceeded by sending a picture of every layer to the company. Every picture was accompanied with text describing the layers and their content. Once the company examined the revised models, we were able to meet with development engineer online via a video meeting. The engineer then explained that the API will be used by a part of their organizations in the near future, however, in the distant future, Tetra Pak's customer will also be using the API. This meant that Tetra PaK is interested in observing the change in the the API ecosystem through time. For this reason, we needed to make a custom model for every important time frame. We decided, with the cooperation of the supervisor and the participant, that a model must be created that represent each of the following timelines:

- Present View of the API ecosystem (No API);
- Near-Near Future of the API ecosystem (The Market Companies partially uses the API);
- Near Future of the API ecosystem (API used fully by the Market Companies);
- Distant Future (API used by the customer and the Market Companies).

This resulted in the creation of four models of the same API ecosystem showing the transition of the company, the four models can be found in the Appendix in Fig. 26, 27, 28 and 29. Focusing on one time frame at a time allowed us to assign a layer to every actor, hence we were able to represent the relevant API layers in each model. The transition from non-layered to layered model can be seen in Fig. 9 which shows a simplified version of the API ecosystem. For a complete version please refer to the Appendix Fig. 30).

After mapping the four different time frames of the API ecosystem, we were able to highlighted the areas that will change through time in order to better visualize the area of interest for the company.

We experimented with the coloring the elements causing the problems that is to be fixed by the API with the color red and teal for elements that eliminate the red elements to show the elements that will be affected by the API. As we advance in the snapshots of the API ecosystem we observe that the teal elements are taking over the red elements showing a clear transition of the API. This can be seen in Fig 10, which shows a simplified version of the first and last transition face of the API. For the detailed models please refer to Fig. 31, 32, 33 and 34 in the Appendix.

As seen in Fig. 10, the company currently lacks an API and the data cannot be easily accessed. This makes the customer rely heavily on Tetra Pak's Market Companies which in turn rely on Tetra Pak Development to find the data. In the distant future, the API will be access by Tetra Pak Development, Tetra Pak's Market Companies and the Customer. This will allow customers to generate reports themselves, or the market companies can capitalize on providing high quality reports to customers by using the API.

In parallel, we could already extract certain design implications and questions that could not be otherwise seen in plain text, such as user-stories and use-case documents, this will be explained in more details in the Discussion section. Modeling the API ecosystems using i* framework also revealed certain limitations that will also be explained in the Discussion section.

2) Modeling Axis' API Ecosystem: Axis Communications shared a single use-case that was approximately two months after Tetra Pak, this means that little modeling could be done, comparing to Tetra Pak, prior the interactive workshop, hence not as many models were generated.

The use-case describes a third party License Plate Recognition (LPR) service that analyzes video footage and is running in the Axis cloud. A parking lot administrator can use the said service in order to manage a parking lot. The API aims to serve both parties: the parking lot administrator (customer) in accessing the LPR service and the third party software developer (LPR service in this case) in integrating the service with the Axis cloud. Hence, the API will have two main users.

The API ecosystem of Axis Communications is quite different from Tetra Pak. From modeling the previous company we learned that organizations will focus on different aspects of the API ecosystem and therefore some modeling criteria will be different. For instance, Tetra Pak was more concerned with the transition to an API, hence the development of four different models, while Axis tends to focus more on the different users of the APIs rather than the transition.

Prior the interactive workshop we formed a model using the use-case provided, which resulted in Fig. 11 that was fairly simple given the shortness of the use-case. Give the lack of time and the shortness of the data provided, we could not identify as many gaps as we could with Tetra Pak, however in the interactive workshop, Axis Communications exposed more information about their API ecosystem which led to the growth of the API ecosystem model as seen in Fig. 38 in the Appendix. In this figure, there are actors that are unconnected

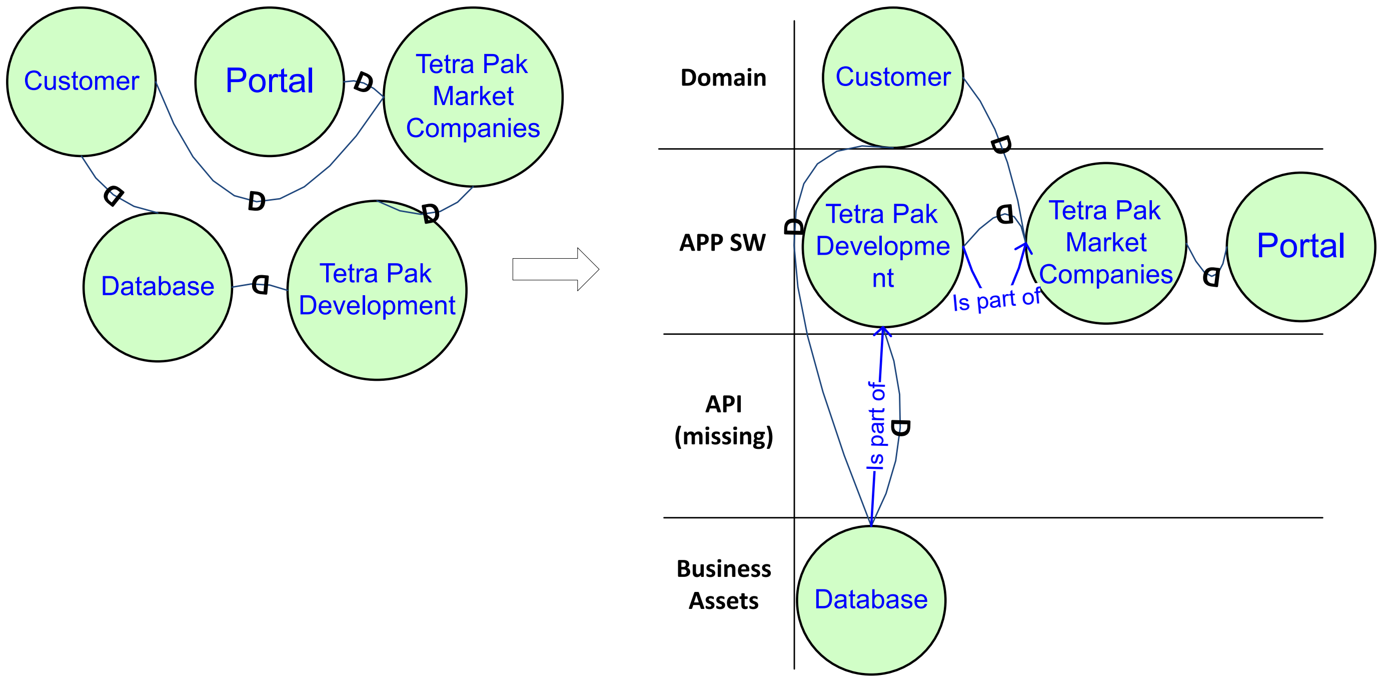

Fig. 9. Before and After Applying relevant API layers to the API ecosystem models - Simplified Asis Model

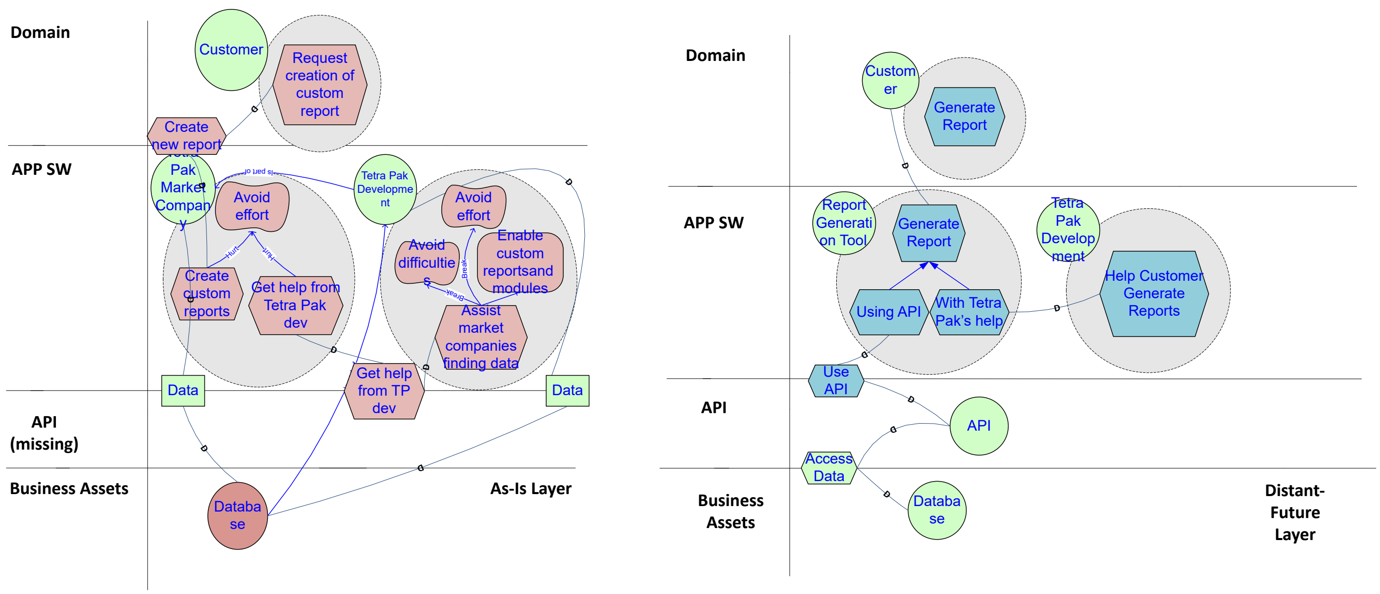

Fig. 10. Simplified Version of Current and Distant Future Perspective of Tetra Pak's API Ecosystem

and do not have any specific goal, just as with Tetra Pak's interactive workshop, not everything could be modeled during the meeting, hence the completion of the model took place after the workshop.

Another difference between Tetra Pak and Axis Communications is that the latter already has an API that is already being used. The company is looking to create another API to be used by different users in their API ecosystem. The company sells network cameras that uses an existing API to access certain data from the camera. The company also plans to create a Cloud API that allows third party software services to be integrated in the Axis' cloud and also allow their customers to use such services from the cloud interface, hence the sub-functions of the API are split into two parts: Content API and Video API respectively. For the company, it is ideally that one API is developed to serve both users, third parties and the customers (instead of creating a separate API for each). Hence the cloud API will have two configurations.

This means that the API ecosystem of the company contain many different level of APIs: The pre-existing API, the Con-

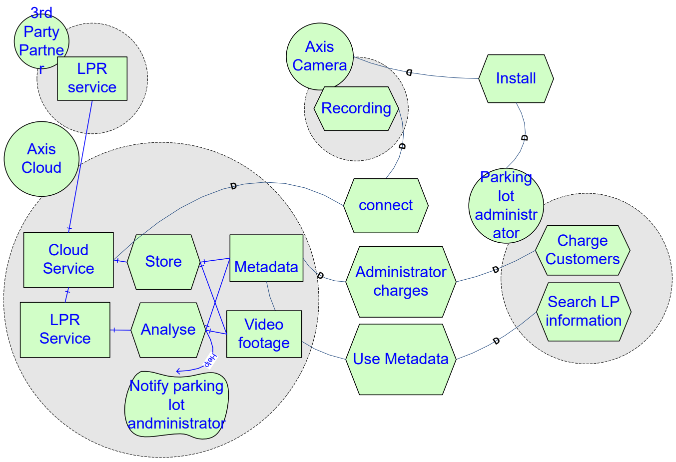

Fig. 11. API ecosystem based solely on the use-case

tent API and the Video API. This nesting can be seen in Fig. 39 in the Appendix.

When it comes to applying the relevant API layers to the ecosystem with more than one API configuration, the actors changes layers depending on which perspective you examine the API. For instance, if you look at the ecosystem from the pre-existing API the Business Asset exposed is the video footage, however if you are targeting the API that is used by third party software (Content API), then the business asset is the Axis cloud. This means that, just like Tetra Pak, there was a need to create a model for every perspective, but in this case the perspective was by differing the API configurations, not by time. A simplified figure (Fig. 12) show the different placements of the actors depending on the API perspective (Content API and Video API).

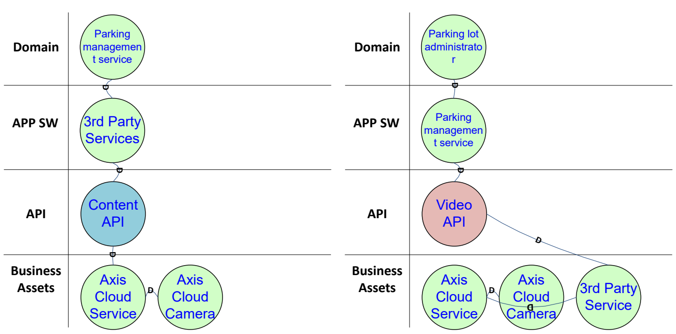

Fig. 12. Illustaration of Changing Actors in an API Ecosystem Depending on API Users

Every perspective is covered within a separate tab in Microsoft Visio. This means that four tabs were added in addition to the complete model;

- Existing API tab: This tab lists the existing API in the API layer,
- Cloud API tab: containing both Content and Video API,
- Content API tab: This tab places the actors from the perspective of the API used by third party software,
- Video API tab: This places the actors on the layers from the perspective of the API used by the customer.

We can summarize our final versions of both API ecosystems in Table I:

TABLE I
FINAL VERSIONS OF API ECOSYSTEM MODELS

Companies Area of Interest	Tetra Pak Time line of the API	Axis Commuication Users of API
Model Generated #1	As-is Fig. 31	Pre-existing API Fig. 43
Model Generated #2	Near-near-future Fig. 32	Cloud API Fig. 44
Model Generated #3	Near-future Fig. 33	Content API Fig. 45
Model Generated #4	Distant-future Fig. 34	Video API Fig. 46

B. Benefits and Drawbacks of Modeling API ecosystems

The introductory group interview as well as the interactive workshop was necessary to understand the companies' API ecosystems, but in order to answer the second research question, we conduct a final online interview with each company once the modeling process has finished in order to discuss the benefits and drawbacks of API Ecosystem modeling. All the data collected from these interview can be found in the Appendix under: Online Interview Data - Tetra Pak and Online Interview Data - Axis Communications.

Company	Variable	Databit	Assigned Level
Tetra Pak	Usefulness of layering the models	"Good to have some kind of explanation" - "see what's happening from layer to layer"	High
Axis		"Give you smaller pieces to get a grip on"	High
Tetra Pak	Usefulness of applying colors	"It [the colors] makes it easy to follow the step you take."	High
Axis		"For now it doesn't add anything for me. But if the next step"	High
Axis		"when I do the model again [in the future] then the colors I think will be really useful"	Low
Tetra Pak	Complexity of the models	"I think it is good that there is all the information here … You need the full picture" - "you can kind of take out only the red and blue parts for communication purposes"	High
Axis		"whatever comes up in those discussions needs to go in there. Because some of them will prove that you have some connections that might show up to be problematic." - "you can have some filtering mechanism or different view where you can take away some of it"	High
Axis		"[Without the layering], it's just a big spiderweb that you look at and think that it is too much."	Low
Tetra Pak	Complexity of the modeling language	"It is hard to understand for people who don't know goal modeling"	Low
Axis		"it [depends] if you only need to understand the model or you are the one actually drawing the model"	Medium
Tetra Pak		"Also a document describing the layers must be included"	Low
Tetra Pak		"It would be good to have a really easy example … or an appendix explaining the syntax that is distributed with all the model"	Low

Fig. 13. Thematic table of qualitative data.

1) Applying Layers to the API Ecosystem: Both organizations expressed positive feedback when it comes to assign the actors to the four discussed layers as it breaks the complex model into four units. Additionally, layering the model also clearly states which actor belong to which layer, hence adding further information into the model.

However, the limitation in using layered approach is that the reader must understand the purpose of each layer. Therefore, a minimal training must take place in order to take advantage of relevant API layers in API ecosystem models.

2) Highlighting Area of Interests: To highlight the area of interest, we colored the elements representing these areas for both companies. This technique proved more helpful for Tetra Pak as the color used were descriptive (red elements represented the problematic elements of the model and teal color showed the solution that is to be implemented to fix the said problem).

In Axis Communications case, the colors where used to indicate the different API variants in the models, this was less helpful as the elements' text already describe the name of each API variant, making the highlighting redundant. This became clean when a participant stated *"For now it doesn't add anything for me. But if the next step, when I do the model again with the new knowledge that we have, and then the colors I think will be really useful."* suggesting that coloring may be helpful in future steps.

3) Complexity of API ecosystem models: When it comes to the complexity of the API ecosystem, both company provided similar feedback stating that it is both a strength and a weakness at the same time.

Furthermore, when asked about the knowledge necessary to understand such models, one participant explained that only the syntax of goal modeling is mandatory to understand mappings. However, another participant added that it depends on the level of involvement of the participant during the modeling section. For example, if the participant is only needed to answer the questions, then a simple introduction to the syntax is sufficient. Although, if the participant is asked to evaluate to correctness of the models, then further training is required. In summary, a participant stated *"I wouldn't say I feel that is any harder than some other modeling languages."*.

VII. DISCUSSION

A. RQ.1: Applying Goal Modeling to API Ecosystems

1) Data Sources and Initial Modeling: To start model, the researchers needs to collect data in order to understand the API ecosystem and model accordingly. The amount of information received from the companies differs greatly. Tetra Pak offered two user-stories that has around 40 pages combined while Axis communications provided one use-case explained in a single PowerPoint slide. This had a direct effect on the completeness of the initial models we created prior to the interactive workshop.

In our case, the increased amount of information received from Tetra Pak enabled us to create a larger model and to identify more gaps in the API ecosystem. Nevertheless, it is important to state that these user-story documents had overlapping information as well as information that is outside of the scope of the API ecosystem such as the process of setting up manufacturing plants.

It is difficult to conclude how much data must be gathered for the initial modeling, or what document will be the best fit for the mapping process. However, we believe that one user-story or use-case is sufficient to make the initial model comprehensive and identify some gaps which is necessary to direct the interactive workshop. In addition, the user-story / use-case must provide enough information to fully grasp the relationships and links between the actors.

2) Cooperation with the Companies: Once a preliminary model is formed and gaps has been identified, the companies' cooperation is necessary to evaluate the correctness of the API ecosystem model. From our experience, companies' cooperation is scarce due to time and resource constraints, also, the participants must be introduced to the basics of goal modeling and relevant API layers.

Once the participants are familiar with the modeling process, the identified gaps must be discussed to fill any incompleteness of the ecosystem. It is difficult to determine when the model should be considered complete. This is because as the companies cooperate in the modeling process, new gaps are identified and the modeling continues iteratively. However, when the participants were asked about completeness of the models created, a Tetra Pak participants stated *"There is all the information here, but it is a little bit messy"*.

Applying goal modeling in an industrial setting revealed a more complex procedure than modeling an ecosystem of a fictitious example such as Yu's application [8]. The process requires the cooperation from the company and this can get difficult to control. In the interactive workshop, every participants adds their own insight and perspective of each actor, and the API ecosystem can quickly drift away from the original scope. This is apparent during the Axis Communications workshop where questions like "Is this relevant to the API ecosystem" or "Is this considered inside the scope of the API ecosystem" were frequently asked.

3) Identifying Areas of Interest: Conducting this research on two companies allowed us to compare the steps taken for each company. The steps were almost identical in the beginning, the only difference being the documents provided, Tetra Pak shared two user-stories and Axis offered a single use-case. However, both documents were read and analyzed in the same manner as we extracted all actors and their goal modeling elements. This step led to the generation of a series of questions. However, once the ecosystem was modeled, each company expressed interests in different aspects, Tetra Pak focused on time and Axis was concerned with the usage of the API. Once the area of interest is identified, it has to be represented in the API ecosystem model. In our case, the representation came in creating additional API ecosystem models where each model map the API ecosystem from a different perspective. The development of such models can be used to analyze how different strategies affect different actors.

This task is useful, because taking into account the aspect of interest (for example the transitioning for Tetra Pak), new models will emerge, highlighting the aspect. Meaning, the visualization of the area of interest will allow companies to see how the rest of the API ecosystem connects to the said aspect, and can examine different design choices based on the visualization. In this research, identifying the aspect concerned allowed us to break the model into different versions, for instance for Tetra Pak, the division was time-based (Current, Near-near, near and distance future). After this division, all of the actors were belonging only to one layer. This made layering the API ecosystem model much easier.

4) Recommended Methodology for Applying Goal Modeling to API Ecosystems: Goal modeling has not received enough application in real world scenarios, therefore there is currently no methodology established [11]. However, from our modeling process, we can extract eight useful steps that help us create a complete API ecosystem model.

1) Acquire relevant documents from a company by asking for some initial data, like use cases or user-stories.
2) Create a preliminary API ecosystem model based on the documents provided.
3) Introduce the company to the basics of goal modeling and relevant API layers.
4) Improve the generated model with the cooperation of the company and apply relevant API layers to the model.
5) Identify strategic area, (e.g., transitions, multiple API views) in the ecosystem model.
6) Create additional models (and / or focused-views) representing different states of the strategic area (if necessary).
7) Confirm the model's correctness with the company, for example through meetings.
8) Explore strategic decisions from the API ecosystem model. For instance by considering the alternative scenarios in the models. (Discussing the OR links).

It is also important to reflect on the failed attempt at applying layers to the models. We found it difficult to include relevant API layers to the API ecosystem mapping in the beginning of the process given the limited knowledge acquired by that time. Additionally, we needed to make sure there is no variables in the API ecosystem that will cause actors to change layers, and if there are such variables, then different models must be created to cover all the scenarios. For this reason, we recommend to apply the layers after the interactive workshop, this is because the researchers will have an adequate amount of knowledge about the API ecosystem by explore the identified gaps with the companies.

With other companies, the steps explained above may change slightly depending on the documents shared by the organization. However, the process will remain the same. Initial models will be generated, which in turn will raise questions regarding understand the API ecosystem. Once this iterative process ends, and the API ecosystem mapping is satisfactory for the company, the researchers must identify area of interest, or the aspects that concerns the company the most, in terms of applying design decisions and taking strategic steps towards the API. This led to the creation of additional API ecosystem models, each model represent an important perspective, that gives strategic value to the company.

Additionally, we made several attempts finding the best way to represent the four layers in the model, however, from our findings, we can conclude that visualizing the layers in the model is the easiest way for the researchers, and it is also convenient for the companies. Moreover, we attempted to layer the model already from the start, but found that it is better to apply the layering once the API is properly modeled. This is because in the beginning of the modeling process, not enough information may be collected to conclude what actor belong to which layer. Furthermore, we do recommend analyzing every layer on its own in relation to the entire API ecosystem in addition to analyzing the connections between the layers. This can be done by creating models that only contain one layer, as in Fig. 22, 23, 24 and 25.

B. RQ2: Benefits and Drawbacks of Modeling API ecosystems

While the modeling process and the interactive workshop helped us answering the first research questions, the online group interviews, alone with the modeling of the API ecosystems, helped us pinpoint the main advantages and disadvantages of modeling an API ecosystem, targeting the second research question.

In this section, additional comments from the participants will be used to reflect on the usefulness of applying goal models to API ecosystem, hence identifying benefits and drawbacks.

1) RQ2.1: Benefits of API Ecosystem Mapping:

*a) **Detailed Visualization of API ecosystems**:* One of the main benefits of modeling an API ecosystem is that it provides a detailed visualization of the entire API ecosystem in one model rather than long textual representation, a participant stressed *"It is big advantages that this is an image and not text, but it is good if there is some kind of workshop and brainstorm and everyone knows the syntax and then build them together. I see a lot of advantages with this"*. It also displays the actors that are involved in the API ecosystem as well as what data is exposed to them. As one of the participants from Tetra Pak stated *"benefits and strength is definitely is giving the overview understanding where the level in which you are working and to whom you are exposing it and in this case the strategy in the steps you like to take and when the different actors get to different benefits and you understand the journey you will take ... It is good that it combines all the elements so you can see the connection in the big picture"*.

Moreover, gaining all this knowledge without a mapping the API ecosystem would necessitate gathering a reading a lot of documentation. Additionally, explaining the API ecosystem without a model would require a significant amount of writing, as the same participant added *"Once you get the goal modeling syntax, you understand all of this, otherwise you will have to use and write a lot of text to explain the same thing"*.

More specifically, Tetra Pak participants found the colors helping in highlighting the transition of the API, however, other means of highlighting may be used depending on the best way to accentuate the area of interest for the companies.

*b) **Focused-View**:* Another benefit, suggested by Tetra Pak's participants is creating a focused-view model that only contained the colored elements (area of interest) in order to discuss possible strategies and design decisions from these elements, *"potentially for specific purposes if you were to communicate today what the near near future, near future and distant future you can kind of take out only the red and blue parts for communication purposes"*. This benefit was also shared by Axis, as they created a focused-view where they only extracted the APP SW layer and the Business Asset layer, the model is in the Appendix under Fig. 47 as a participant said *"you can have some filtering mechanism or different view where you can take away some of it. So the hurt, break stuff maybe you don't need that to see the relationships, that more of a what we want to do and how they impact each other."*. Moreover, the participant showed the progressing of data generation in the focused view. In the figure, we can see that the video service uses the Network Camera as an asset to produce the Video, that will be used by the Motion Detection Service to produce the Motion Data as asset and so on. Therefore, this focused-view can be used to communicated how the business assets are produced and used from a service to another.

This focused-view model may be useful to mitigate the limitation of goal modeling discussed by Sadi and Yu [11]. They both discussed goal model's inability to represent different views of a SECO, however, strategically extracting portions of interest of the API ecosystem model allows goal modeling to have such functionality.

Nonetheless, the larger context of the entire API ecosystem must be considered, therefore having a complete model is needed to accurately represent the API ecosystem as the same participant explained *"I think it is good I have the complete picture ... Hiding the green ones when you present the strategy to taking it step wise and then when it is more like documentation of the total, then you have all of it"*. Moreover, these focused-views are based on the API ecosystem model, hence mapping the entire API ecosystem is needed to extract the focused-views. Moreover, elements outside the area of interest are also important as they need to be considered when designing the API, as an Axis interviewee said *"I think that whatever comes up in those discussions needs to go in there. Because some of them will prove that you have some connections that might show up to be problematic."*.

*c) **Layering the API ecosystem Model**:* Layering the model was also helpful in separating the actors, this not only indicates to which layer all the actors belong to, it also improves the organizations of the model, which can easily seem more cluttered, when the participants were asked about the layers, both participants from Tetra Pak replied *"It does help, you separate the business assets and the API. that is especially helpful"* and *"the layer helps make it a little bit clearer ... but it is good that there is layers"*. Similarly, an Axis Participant added *"It is much more useful to make it layered because then so you have to think a bit about what is what here. Otherwise it just gets, it's just a big spiderweb that you look at and think that it is too much. I think it is really useful to make it layered, partly for the process is useful, but it gives you smaller pieces to dive into that you actually can get a grip on"*.

*d) **Reveal Gaps in Knowledge about API ecosystems**:* Another benefit of API ecosystem mapping is the fact that it exposes gaps in the API ecosystem that is otherwise hidden in textual files. Modeling the API ecosystems with the companies also helped triggering discussions that revealed information that was not available in the documents received. Modeling all these information in one model gives the companies the opportunity to rely on one model to have a complete high-level view of the PI ecosystem.

*e) **Analytical Advantages**:* Having a visual representation of the API ecosystem easily shows all relevant actors and how these actors depend on each other. The model also shows how the goals are accomplished. Considering this aspect of goal modeling, one can analyze any activity and consider alternative configurations and see how relevant actors and elements are effected. As one of Axis participant added *"As a working process I think it was really good. My feeling is, by doing this you will find some configurations that are more pleasing than other configurations."* when asked about generated different models for different scenarios.

2) RQ2.2: Drawbacks of API Ecosystem Mapping:

*a) **Resource Extensive**:* API ecosystem modeling is a resource intensive task. In order to create a syntactically correct model, the researchers must be familiar with the i* framework.

Additionally, the modeling process requires constant contact with the company which can be difficult and time consuming. In our case, we needed to conduct a group meeting with the companies before the modeling process started, this was done so the companies could explain their API ecosystem and to provide relevant documentations. We were able to create the initial mapping of the companies API ecosystems without the companies present of the companies. However, after every important iteration, there is always a need to confirm the correctness of the API ecosystem regarding both, its syntax and its context. In total, the API ecosystem models went through around 40 models to reach its final stage and all of the modeling must be supervised by a goal-modeling expert.

*b) **Model Complexity**:* When the API ecosystem model is complete, the model is usually too complex to understand, and the stakeholders always needs a reminder of goal modeling syntax as well as brief explanation of every layer. As a one of the participant from Tetra Pak explained *"it is hard to understand for people who don't know goal modeling"* and another participant added *"One weakness is it becomes messy and complex and I don't know in terms of complexity what the upper limit is"*. Moreover, with every iteration sent to the companies, they need to be reminded of the syntax of goal modeling and an explanation of every layer used, *"The last time we looked at it together, we had layer description for each* *layer last time ... for those who hasn't been in discussion of the layers, it would be hard to discuss the business assets and APP SW as separate layers I think it's good to have an explanation of each layer ... the syntax would be good to have a really easy example ... or like an appendix or a separate document explaining the syntax of modeling distributed with the model"*.

Presenting a sample example of a goal model was done in the beginning of the interactive workshop, however, according the data collected, this example must accompany every iteration sent to the companies and serves as a reminder of how to use the i* framework.

*c) **Learning Curve**:* This lack of knowledge also hinders the participant to add additional layers if needed *"I haven't used the models so much, but it is good that there is layers, but I'm not used to the model and I can't say if these are the right layers or not"*. In other terms, if the participants do not posses enough knowledge about goal modeling and API ecosystems, they may not be able to add necessary layers to the model, making the mapping of the API ecosystem incomplete.

The need to include a document that reminds the stakeholders of the syntax of goal modeling is expected, as goal modeling lacks textual representation [11], however, there is also a need to include an explanation and description of all layers used in mapping. This adds to the level of knowledge required to understand these API ecosystem goal models.

*d) **Inability to Display Degree of Importance**:* Another limitation of goal modeling is that it did not allow us to show the degree of importance of such elements in an API ecosystem. Some elements were relevant to the API and was worth considering and mapping however they were of lower importance comparing to other elements. For instance, inside the Portal actor in the Tetra Pak ecosystem model, the module was said to be of low importance comparing to the dashboard or the reports. This was difficult to illustrate in the model, if this was possible, one can easily know which elements to prioritize when making design decisions.

Below we provide a summary of benefits and drawbacks of goal modeling API ecosystems:

Benefits

- Detailed visualization of API ecosystem in one place.
- Clear display of data exposure within the API ecosystem.
- Easily follow flow of information.
- Ability to highlight areas of interests.
- Applying layers makes the model easier to read.
- Applying goal modeling helps reveal gaps.

Drawbacks

- Modeling takes a long time.
- Constant need to remind companies of goal modeling syntax and relevant API layers.
- Models can become complex to read.
- Requires i* framework expert.
- Requires companies cooperation.
- Unable to show degree of importance of elements.

VIII. Conclusions & Future Work

This paper serves as a starting point to API ecosystem mapping using goal modeling. Given the complexity of the API ecosystems, the modeling can be challenging. We found that both the models generated as well as the modeling process can benefit a company. The modeling process elicits a lot of discussions about gaps identified that would have been difficult to bring up otherwise. Additionally, once the model is complete, an organization can easily extract focused-views for communications purposes. This means that every generated focused-view will serve a specific purpose. Nonetheless, some participants hoped for an easier way to switch between models and focused-views of the API ecosystem, as well as an easier method to derive focused-views from the main model.

Application of goal modeling on real life cases are scarce in the research community, and given the resource limitation, we were only able to map the API ecosystem of two companies. Further application of goal modeling is needed to confirm the modeling steps provided in this study.

A future direction proposed by the Axis Communications is to apply Value Network Modeling [14] in order to assess the business value provided by the actors. Moreover, experimenting with tools and methods to facilitate the creation of focused-views can speed up the modeling process.

IX. Appendix

A. Interview Data

1) Tetra Pak Interactive Workshop Schedule: Introduction (20 minutes)

- Introduce background work on Ecosystems (from past work).
- Introduction to goal modeling as a potential Ecosystems modeling language

Interactive Modeling for user-story Number 1 CIP Reports (60 minutes)

- Use incomplete starting model prepared before hand
- Researchers go through user-story confirming their understanding of the ecosystem while Supervisors models the discussion, showing on a projector
- Participants view and correct/expand model

Break (15 minutes)

Interactive Modeling for user-story Number 2 Quality Control System (60 minutes)

- Could use elements of ecosystem for user-story Number 1
- Researchers go through user-story confirming their understanding of the ecosystem while the supervisor models the discussion, showing on a projector
- Participants view and correct/expand model

Conclusion (20 minutes)

- Discussion of process (what went well, challenges)
- Discussion of way forward (further modeling or analysis)

2) Sample of Tetra Pak Interactive Workshop Interview Guide: List of questions extracted from initial modeling of the user-stories:

1) The content of the portal, where is this saved? Tetra Pak's database or customer's database?
2) Can this be considered as the host of the end-product? (reports) if so, shouldn't it belong to the domain?
3) The portal does not seem to belong to any layer. Maybe we need a new layer?
4) Is the module part of Portal or how does it fit in the ecosystem?
5) Where is the module used in a real life scenario? Is it used through the portal?
6) How does the TPM is related to the CIP process? Does this mean that whoever will do the CIP configuration must know how to operate the PLC to get (sensor) data?
7) Who are the expert Exactly? Are they from Tetra Pak or from the customers or?
8) The explanation of the CIP Summary Report is missing. (Says See also chapter xxx)
9) What is the difference between Cleaning time interval report and CIP Analyze Time report? Is the former only local in china? Meaning only one exist at a certain point (if in china, then only Cleaning time interval report is generated).
10) To which actor should CIP Reports belong to? Does this hold true to other user stories such as the QC System? Or it depends on the user story?
11) In 3.6 "Relation overview between CIP reports" what does the error mean?
12) Unsure if QC module should be in the API layer or domain layer
13) Should generic solutions be part of the quality control module?
14) What is exactly LIMS system? (Laboratory information management system?)
15) Would the "Tetra Pak" actor considered as Business Assets?
16) Are experts considered a part of Tetra Pak or some external assets?
17) What impact does HACCP and GMP have on the management?
18) How does the TPM work as a standard solution?

3) Tetra Pak Interactive Workshop Data:

- **Supervisors** - “Should we talk about the rest, overview of the model or talk about the detail of this actor?” Tetra Pak - “Maybe it will be good to have an overview”
- **Participant** - “What we are really interesting in right now is how to show this data that is collected, so this is interesting in the big part but for the API report it is not so interesting really how the CIP in the factory is functioning. The interesting part is to get out what the different users want to get out of the API.”
- **Researchers** - “So standard CIP reports are done by Tetra Pak themself or by market companies?” Tetra Pak - “It is done by us here the development department (Tetra Pak)”.
- **Participant** - “We also hope that it will be easier for the market companies to do the reports so it will be more logical”
- **Participant** - “The share will be more like andriod app store some common marketplace to share reports”
- **Participant** - “Isn’t building the standard report the same thing as the create customize report?”
- **Participant** - “First we said we need to start from the top and take some aerial thing that will do with the first custom report and what they would like to see. Then we kind of got stuck in discussions so we didn’t get anywhere so we thought maybe we start from bottom up and see what kind of data we collect and start from that.”
- **Researchers** - “Will the database be modeled as an actor or a resource?” Supervisors - “I believe the database should be modeled as an actor because it is more complicated that way since it can do action and other commands”
- **Participant** - “I dont think its so interesting really, go down to the database again. I mean this is how the database used to look in the old product and it will change over time”.
- **Participant** - “The interesting part would be how the API would look”.
- **Participant** - “Going into the future is how it is stored, that is something that will change in the future and when we find this we need to optimize things and so on, that’s why we need an API”.
- **Supervisors** - “Is this something that is specific to a storyline?” Tetra Pak - “No, This is the general functionality and then everything like quality or specific storylines, this is the boundary where they live within.”
- **Supervisors** - “So instead of working separately on two models I think it’s useful to merge them into one general model” Tetra Pak - “Mhm (Agree)”
- **Participant** - “So we tried to see which is the first Market Company that will use the probably first area then we start to do custom report and I think that is a good approach to see what will be used first. Then we got a little bit overwhelmed, where should we start? We have too much data, and what data should we expose?”
- **Participant** - “We adapt to your needs”
- **Participant** - “it’s not ready rather Its work in progress, that way say that there is a possibility to start use the API but there is only the half way of using the new way to do it. But you can use the old way, the legacy way”
- **Participant** - “The main reason for the API is to customize reports.”
- **Supervisors** - “So the problem is to upgrade custom reports?” Participant - “No the problem is when updating the system, because if we change something like the data structure, then we don’t know if one of those hundred custom reports is breaking”
- **Participant** - “I think it was good that you started to model before the meeting”
- **Participant** - “I think it is a good way of seeing what depends on what”
- **Participant** - “If you put it into layers you can see the benefits for your team and the benefits for the Market Companies and what you like to achieve or support with the API’s.”
- **Participant** - “I can see the potential [of modeling]”

4) Axis Communications Interactive Workshop Schedule: Introduction (20 minutes)

- Introduce background work on Ecosystems (from past work).
- Introduction to goal modeling as a potential Ecosystems modeling language

Interactive Modeling for Top-level Ecosystems (60 minutes)

- Discussion of ecosystem(s), actors and dependencies
- Supervisor models the discussion, showing on a projector
- Participants view and correct/expand model

Break (15 minutes)

Interactive Modeling for Ecosystem APIs in Cloud Setting (60 minutes)

- Start with pre-created model from provided slides
- Researcher go through use-case confirming their understanding of the ecosystem while supervisor models the discussion, showing on a projector
- Participants view and correct/expand model
- Elicitation of possible strategies

Conclusion (20 minutes)

- Discussion of process (what went well, challenges)
- Discussion of way forward (further modeling or analysis)

5) Axis Communication Interactive Workshop Guide:

1) Is the LPR service running entirely in the cloud and has no connection to the third party?
   - If so, then does modeling the third party has any significance in the API ecosystem?
2) What alternative modeling and designing you are considering for API ecosystem design?
3) Additional questions arose during the interactive workshop as the modeling process progressed.

6) Axis Communication Interactive Workshop Data:

- “The data might be able to leave [the cloud], but not in an uncontrolled way”
- “Video is not available there, might be available here because there are no humans”
- “They can have access to the video without actually being given the video”
- ”it’s probably the one [parking lot administrator] determining the rule-set”
- “Maybe we as Axis want to have a billing service that they could use if they don’t have anything cheaper to use”
- “Control how much information you get and it’s an important requirement in volume management”
- “If I want to expand my business, going to an automated system is useful because it probably scales better than hiring more people”
- “Handle exception is the main task for having a human on spot”
- “The exceptions they impact the customer experience”
- “What are interesting patterns and anti patterns that we are trying to find here”
- “For the sake of this model, we say that the business asset is the video in the cloud”
- “In this case the API that we want to look into and strategize is the access to video in the cloud and beneath that becomes the business assets”
- “We want an API that allows somebody to refine what we have and make money out of that so that they in return can pay us”
- “A good API from my point of view is allowing black box interactions”
- “The key point is that the API should hide that there are an axis camera”
- “It doesn’t matter if the video is captured a second ago or ten years ago I just want an API to access it”
- “There are two variance [API’s] of it, I think there is one variant where content stays in this cloud service and one variant that makes it leave”

7) Extra Online Interview Guide: Sample of Questions Asked During Tetra Pak Extra Online Interview:

1) There has been a discussion about adding a cloud solution, how does this relate the API ecosystem?
2) Do you need to call the API in order to manage the report in the portal?
3) When the customer want to generate a new report, do they have to do it through the portal?
4) Where is the portal saved? Tetra Pak's database or customer's database?
5) Does the market company has access to the portal or only the customer?
6) We have talked about letting customer combine data to increase business intelligence, is this considered future feature in the API, and done now manually? "
7) Is taking sampling, and generating sampling results specific to certain user stories or general enough to model in the API ecosystem?
8) In the user stories there is a Module that can generate report, is it correct to say in the future, this model would be replaced with the API?
9) Is the Report Writer actor significant enough to be modeled on its own, or is it possible to include it with customer and market companies?
10) Do market companies only partially help with the report and then the customer must add it to the portal, or can the market companies save the customer's report in the portal?
11) Is the portal considered a hub where all the customers' reports are managed?
12) Will the portal be available for the customer? If so, will the new report end up in the Portal?
13) When the customer or the market company adds a report, do they need to access the portal first?
14) The report manager is modeled as a resource as, to our understanding, is an interface where the customers (and market companies) can access saved reports and view them or delete them. Is this correct?
15) Other than reports, what software will be using the API?
16) This layer is currently in progress, but is this mapping in line with what Tetra Pak has in mind?
17) Is there something that is missing on this layer, that we should take into consideration or add?
18) Are all these functions correct, and are there any other important functions?
19) Is there something that is missing on this layer, that we should take into consideration or add?
20) Are generic solutions for CIP report specific to the CIP user story or is it general to the entire reporting framework?
21) Will the market companies be able to access the database directly in the future?
22) Combining data and showing data in different combinations is considered an API feature?
23) Is the entire process of modeling a plant relevant to the design of the API?
24) Are the sampling plans done by the customer?
25) Do factories use both manual and automatic data or only one or the other?

8) Extra Online Interview Data - Tetra Pak:

- “I just want to make one thing clear there, that is starting to get more and more clear for us is that our main customer will be the market companies and we are aiming more and more for the goal to make products for the market companies”
- “We should think more and more about doing a product that the market company can easily adjust for the end users and that the market companies is our first hand customers.”
- “The portal [component in the API ecosystem] is for displaying reports I would say.“
- “What the portal really does is that it helps you with language, so you can swap between languages, it can help you with purity which makes you protect your different reports depending on who’s logged in.”
- “Market companies that helps the customer combine the data they can reach from our API with any other kind of data [example: lab systems]”
- “The main consumer of the API will be the Market Companies.”
- “We will do some standard reports that will come with the portal as the standard setup. Then the market companies should add more and they could also replace part of the standard reports that we provided”
- “The API will be designed to primarily to feed the reports and dashboards with the data”
- “Maybe in the future it will be the customers that do the reports and add them to the portal themselfs”
- “I think that what we can try to do with the API is to make it simple somehow to combine it with combine data.”
- “The end goal would that it is only us here in the development department [have direct access to the database].”
- “The PLC is not a part of the portal, so we model up in the portal actually what kind of things we want to read from the PLC’s”

9) Online Interview Guide:

1) Questions about layered mapping
   - Are the four layers representative of the entire API ecosystem? Or should there be additional layers?
2) Questions about API view presentation mapping
   - Do you think presenting API ecosystem in different views are appropriate and helpful to deducting design implications?
3) Do you think the colors were helpful to highlight the API focus of each view?
4) Is the amount of information included in the model appropriate to represent the API ecosystem, or is it too much or too little? (if it is too much then does removing the unnecessary ones can make the model more understandable?).
5) Is there another alternative to goal modeling that you think can help map an API ecosystem? How does this alternative compared to goal modeling?
   - If not, what are the weakness that you see with modeling an API ecosystem this way?
   - If not, what are the strengths and benefits that you see with modeling an API ecosystem this way?
6) Can you talk about the learning curve of understanding such mapping? For example, how much training do you think it's necessary for someone in the company to understand this?
7) Do you think these design implications would have been harder to see if the API ecosystem was not mapped?
8) What would you model differently? Or what would you add or remove from the model?
9) Is the fading of the actors in certain models making the API ecosystem mapping clearer? Or is there an alternative way to model this?

10) Online Interview Data - Tetra Pak:

- Do you think presenting API ecosystem in a layered manner is appropriate and helpful to deducting design implications? *"For me, it would be good to have some kind of explanation, what's happening from one layer to another layer."*
- Do you think the colors were helpful to highlight the API focus of each view? *"It [the colors] makes it easy to follow the evolution and the step you take."*
- Is the amount of information included in the model appropriate to represent the API ecosystem, or is it too much or too little? (if it is too much then removing the unnecessary ones can make the model more understandable). *"Domain and APP SW are a little bit unclear, the layer helps make it a little bit clearer but it I'm still struggling sometimes to be able to be separate what is domain what is business assets what is app sw, I haven't used the models so much, but it is good that there is layers, but I'm not used to the model and I can't say if these are the right layers or not, but you are more familiar with the layers and the model than I am"*
- Is there another alternative to goal modeling that you think can help map an API ecosystem? How does this alternative compared to goal modeling? *"No, we talked about system alley? but that is even more complex and varied i guess, so there is no alternative tool or model"*.
  - If not, what are the weakness that you see with modeling an API ecosystem this way? *"It is hard to understand for people who don't know goal modeling" "One weakness is it becomes messy and complex and i don't know in terms of complexity what the upper limit is"*
  - If not, what are the strengths and benefits that you see with modeling an API ecosystem this way? *"Benefits and strength is definitely is giving the overview understanding where the level in which you are working and to whom you are exposing it and in this case the strategy in the steps you like to take and when the different actors get to different benefits and you understand the journey you will take." "Once you get the goal modeling syntax, you understand all of this, otherwise you will have to use and write a lot of text to explain the same thing"*.
- Can you talk about the learning curve of understanding such mapping? For example, how much training do you think it's necessary for someone in the company to understand this? *"The last time we looked at it together, we had layer description for each layer last time ... for those who hasn't been in discussion of the layers, it would be hard to discuss the business assets and app sw as separate layers i think it's good to have an explanation of each layer" "Have the descriptive text separately works just fine, like an introduction"*
- What would you model differently? Or what would you add or remove from the model? *"The actors and portal was in the same place, I was confused when they swapped places." [The actors should be in same places in all tabs]*

11) Online Interview Data - Axis Communications:

- Do you think presenting API ecosystem in a layered manner is appropriate and helpful to deducting design implications? *"It is much more useful to make it layered because then so you have to think a bit about what is what here. Otherwise it just gets, it's just a big spiderweb that you look at and think that it is too much. I think it is really useful to make it layered, partly for the process is useful, but it gives you smaller pieces to dive into that you actually can get a grip on"*
- Are the four layers representative of the entire API ecosystem? Or should there be additional layers? *"My picture is much more oriented towards which API do we want to concentrate on and promote while the model itself is more descriptive of actually how things look and what's needed in more detail. So I think it's more like two views and what would be useful for me would then be to be able to transform between these two views. I think that the boundary objects that you get in the model I think they are kind of a key especially the ones around the API. Because there you will be able to see is this actually an API that I wanna show? Is this an API that is useful for developers? Or is this just some internal component?"*
- Do you think presenting API ecosystem in different views are appropriate and helpful to deducting design implications? *"As a working process I think it was really good. My feeling is but by doing this you will find some configurations that are more pleasing than other configurations. You will see like in a gut feeling way that this way of showing it looks actually better, it feels better and it will probably work better."*
- Do you think the colors were helpful to highlight the API focus of each view? *"For now it doesn't add anything for me. But if the next step, when I do the model again with the new knowledge that we have, and then the colors I think will be really useful. Because then we can more or less immediately identify work packages and areas that we have to prioritize to know more about if it is problematic for example."*
- Is the amount of information included in the model appropriate to represent the API ecosystem, or is it too much or too little? *"I think that whatever comes up in those discussions needs to go in there. Because some of them will prove that you have some connections that might show up to be problematic. For me I would handle this as a brainstorming, just put everything in there then maybe, as I said, you can have some filtering mechanism or different view where you can take away some of it. So the hurt, break stuff maybe you don't need that to see the relationships, that more of a what we want to do and how they impact each other."*
- Is there another alternative to goal modeling that you think can help map an API ecosystem? How does this alternative compared to goal modeling? *"Right know I haven't looked that deeply. I sent a mail [to the supervisor] a while ago and asked for something good to do this and you sent me links to i* language and I looked at it then I thought yeah this can actually work. So I haven't found anything before but I haven't done an extensive search but I did look around a little."*
- If not, what are the weakness that you see with modeling an API ecosystem this way? *"The weaknesses is probably partly connected to the strength that you actually can model even the social aspects of this. You might go overboard in the discussion and in the workshop while you end up concentrating too much on these details and actually miss out on the more technical facts about it. At the same time I don't wanna leave that discussion out. I think the expressiveness makes it more challenging to run the workshop. But at the same time, if you manage to do that it is a positive thing. It is a clear danger when doing these kind of modeling sessions, especially in a group."*
- Can you talk about the learning curve of understanding such mapping? For example, how much training do you think it's necessary for someone in the company to understand this? *"I would take the question into two part. The first is how much you need to understand and know to participate in the modelling session when you are not the one actually drawing the model. Then I think it is quite low in step, I don't think it is that hard to understand how it works and be able to express yourself in a way that the person that is actually drawing it can understand. The other way then is to actually understanding it when looking at it afterwards and trying to understand what the model actually says. There I think it is a bit harder, but I wouldn't say I feel that is any harder than some other modeling languages."*
- What would you model differently? Or what would you add or remove from the model? *"I think that is more or less what I've said before. Trying to reduce the detail in a simple way, that would do it for me"*
- Do you think it will be rewarding to create models based on the timeline of developing the API? *"I think it is a nice approach but I don't think I would do it. I would go for modeling what we want as a end result and I might do the current situation. But in between no, I would leave that to some executing project that works in an agile way to just find the best possible way through there. It depends on the size of the work you are approaching. I mean if it is a five year effort you might want to model some of the steps in between since you might land there for some years."*

Additional quotes:

- "When looking at the pictures of the model made me thinking what we really want from the API"
- "This is the goal that we wanna get to with the API [shows own made model], so this is the strategic goal of the API, that we wanna be able to add new content types and new refinements of the data but we still wanna hide where and how it is produced."

- “I don’t know exactly how to to go on after this but now I know this is the API we wanna concentrate on getting right.”
- “I realized from this process here that there is something here need to take care of in a more generalized way.”
- “We talked about how it was a stacked thing and maybe several ecosystems but I never felt comfortable with that explanation. Because it is one ecosystem”
- “For me the whole activity here have been more of a process in how do you work with these questions and what tools are they and can I get some new revelations by modelling this.”
- “I still think that the way we had to squeeze it to get it into these layers that actually gave us something”

X. Models

1) Tetra Pak:

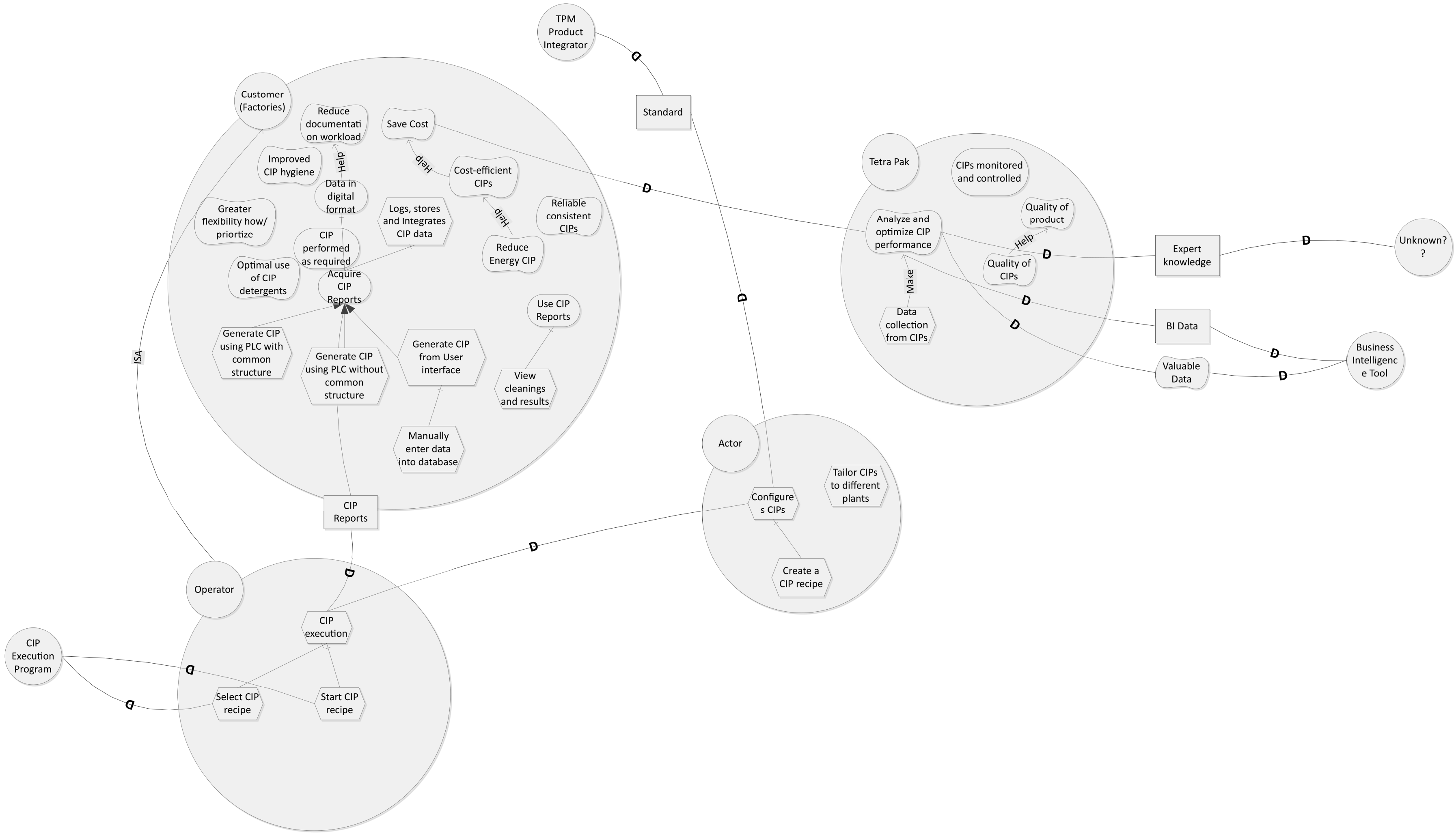

Fig. 14. Initial modeling of first user-story.

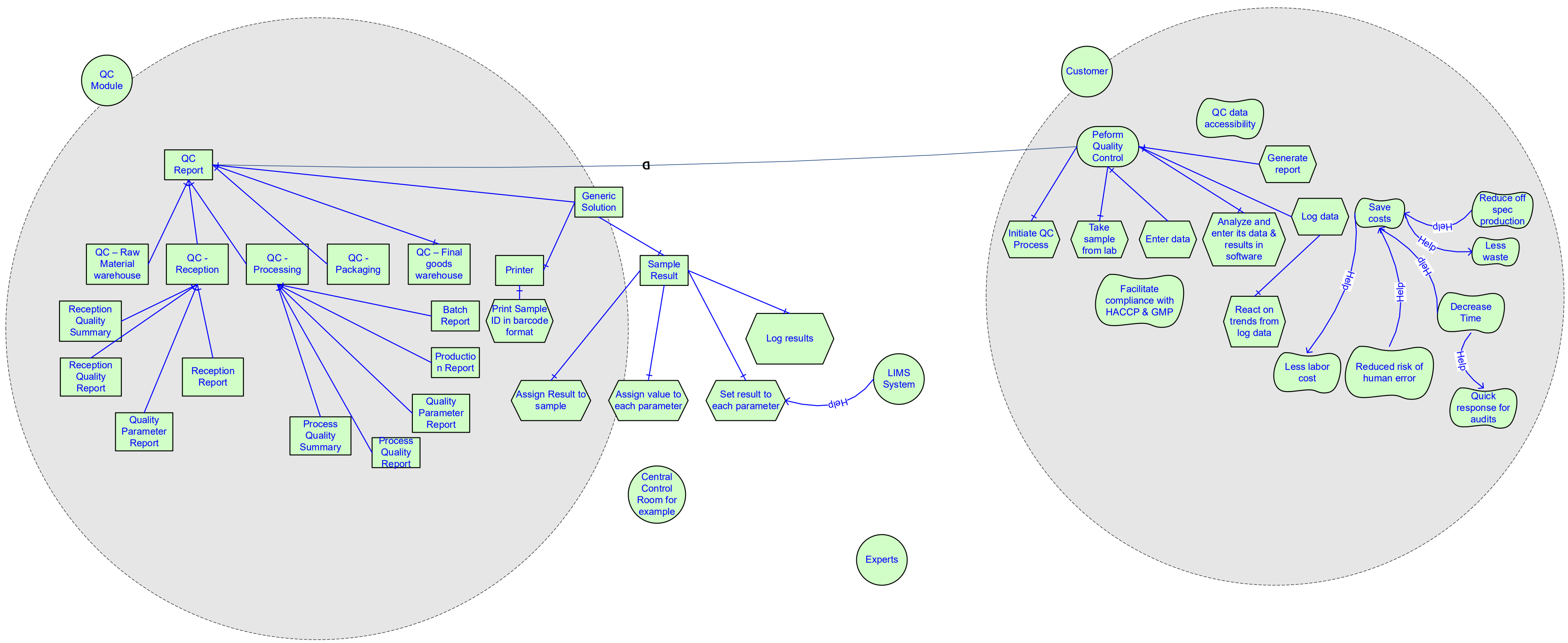

Fig. 15. Version 1 of second user-story.

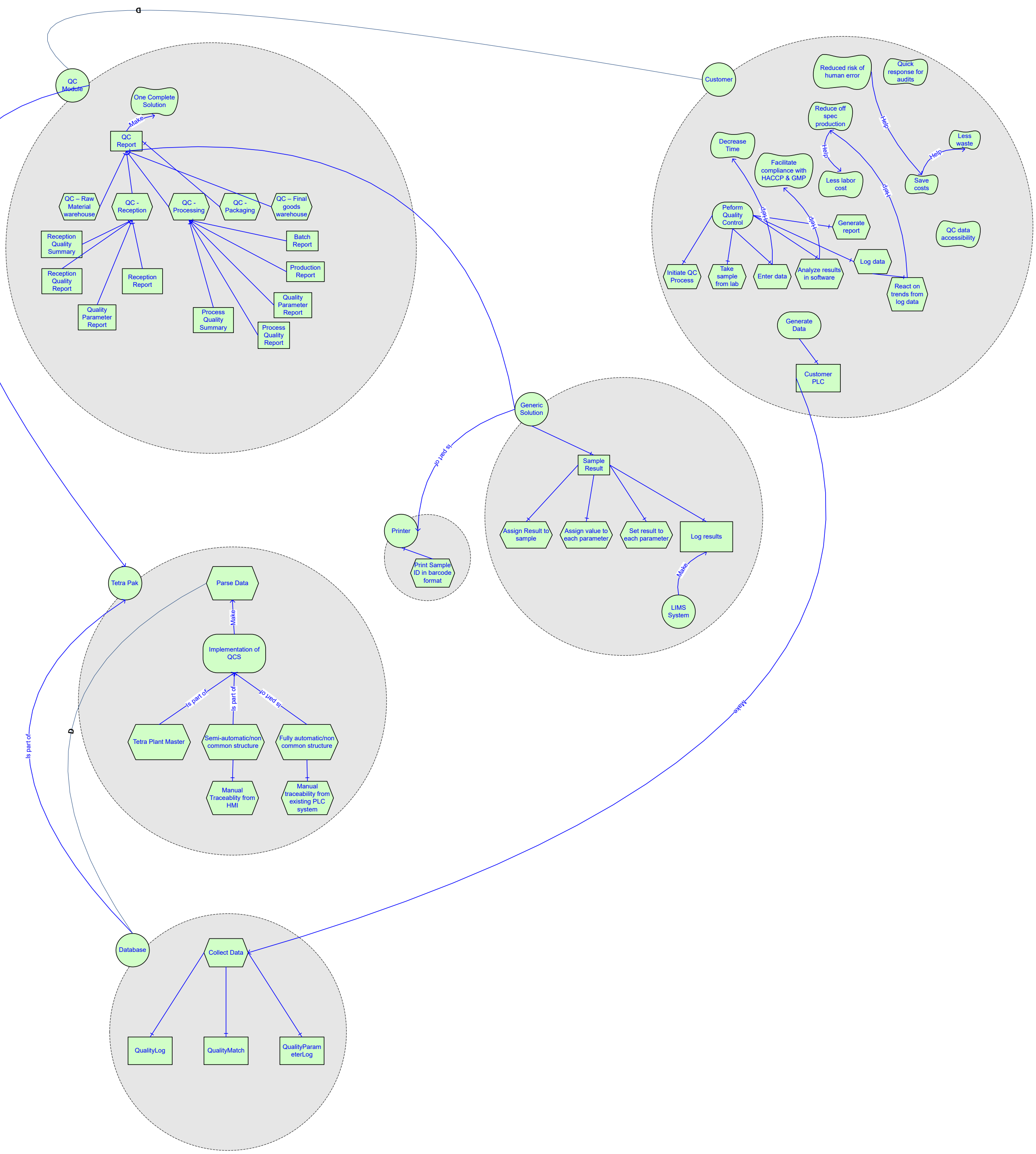

Fig. 16. Version 2 of second user-story.

Fig. 17. Layered version of the first user-story

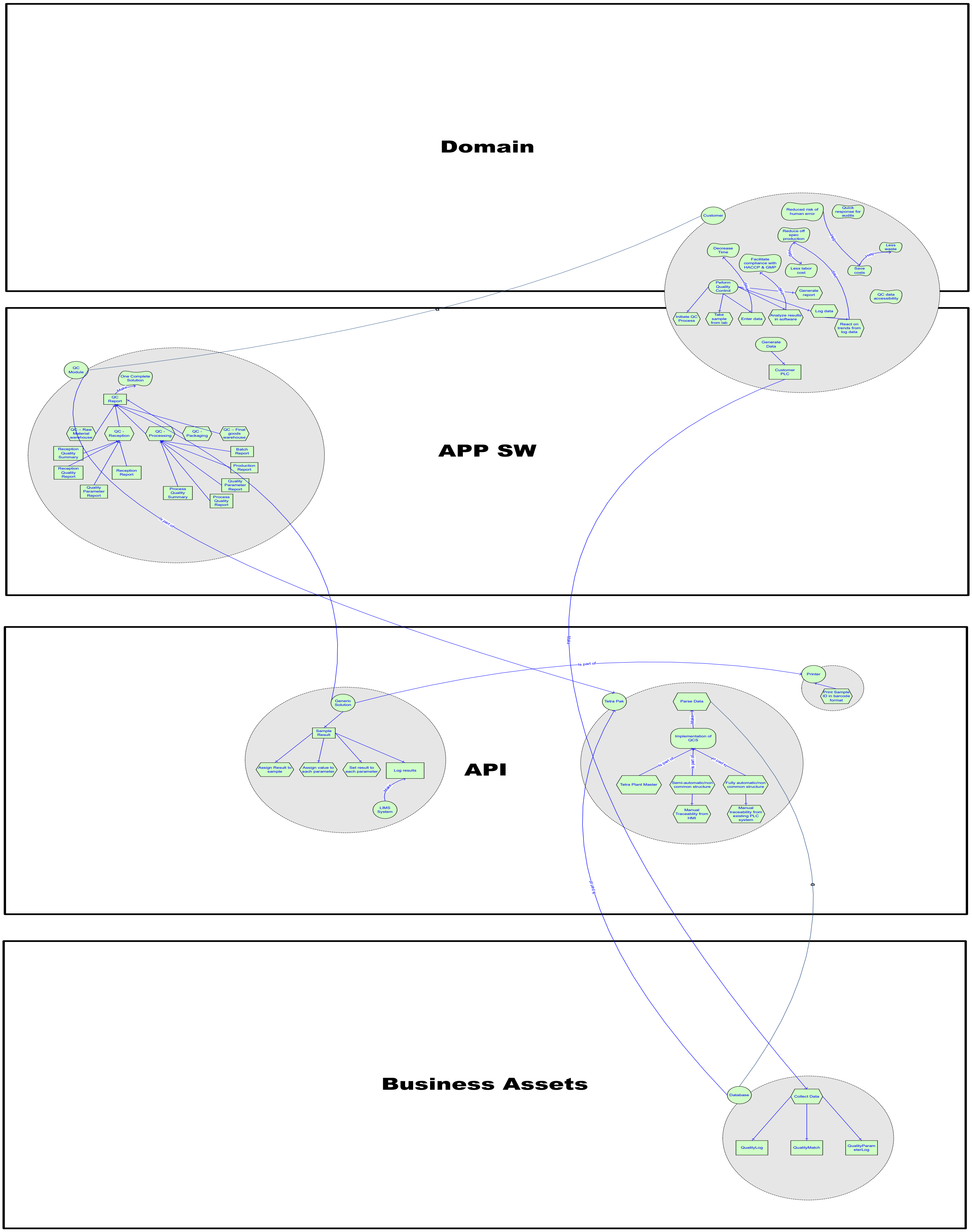

Fig. 18. Layered version of the second user-story

Fig. 19. Version 1 of Tetra Pak API ecosystem - not layered

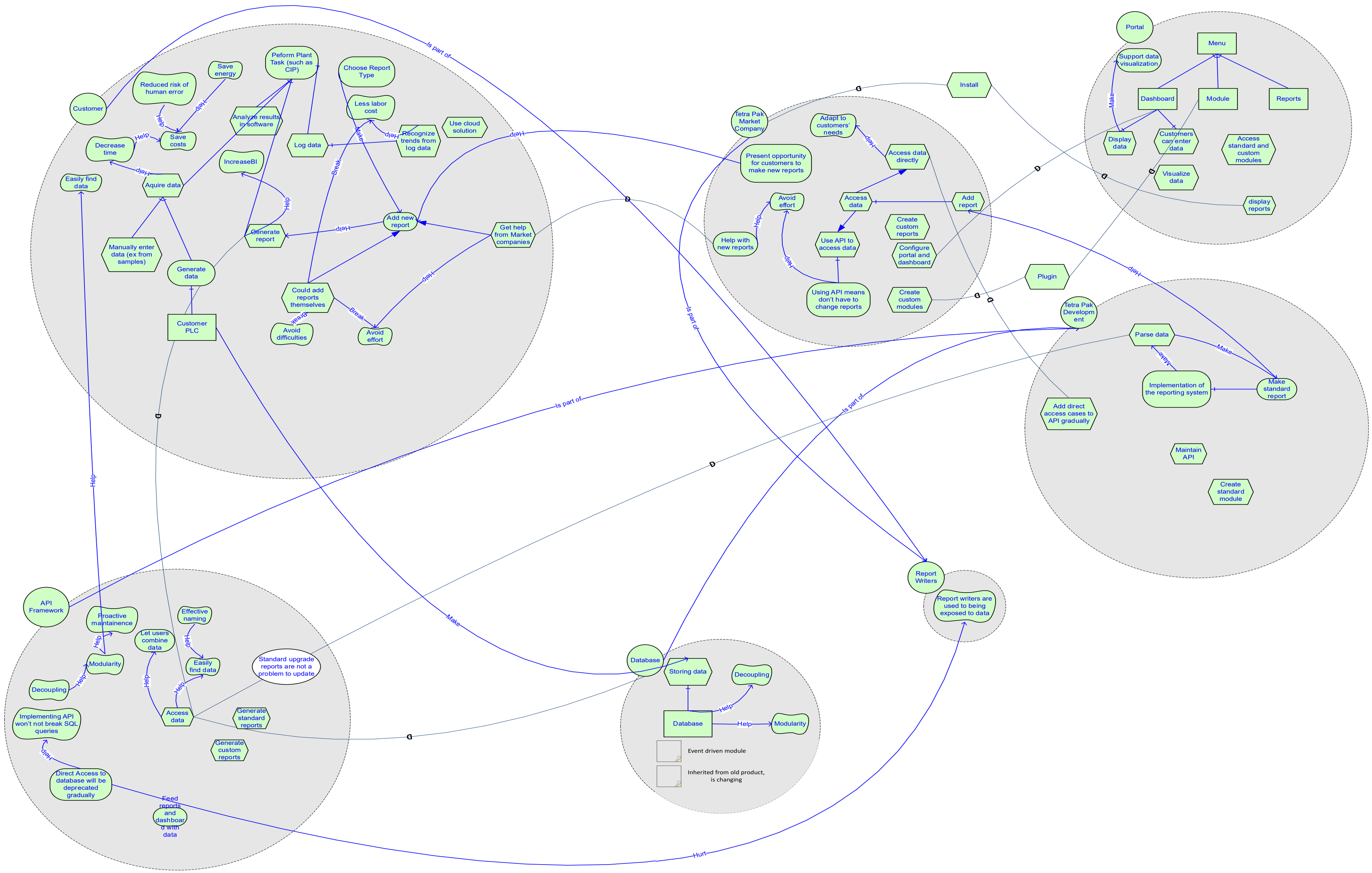

Fig. 20. Version 2 of Tetra Pak API ecosystem - not layered

Fig. 21. Fusing the two user-stories into layered model

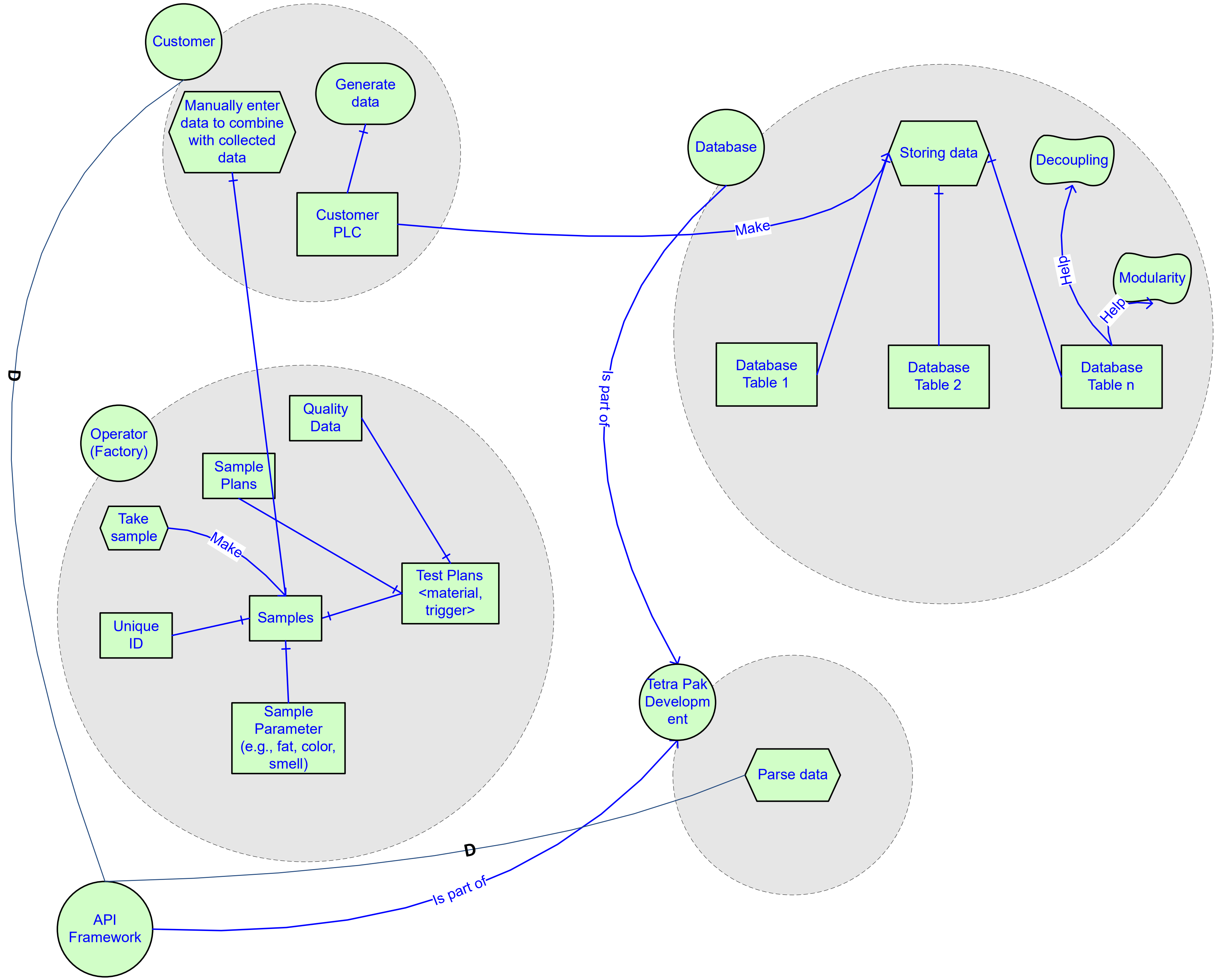

Fig. 22. Domain layer of Tetra Pak API ecosystem

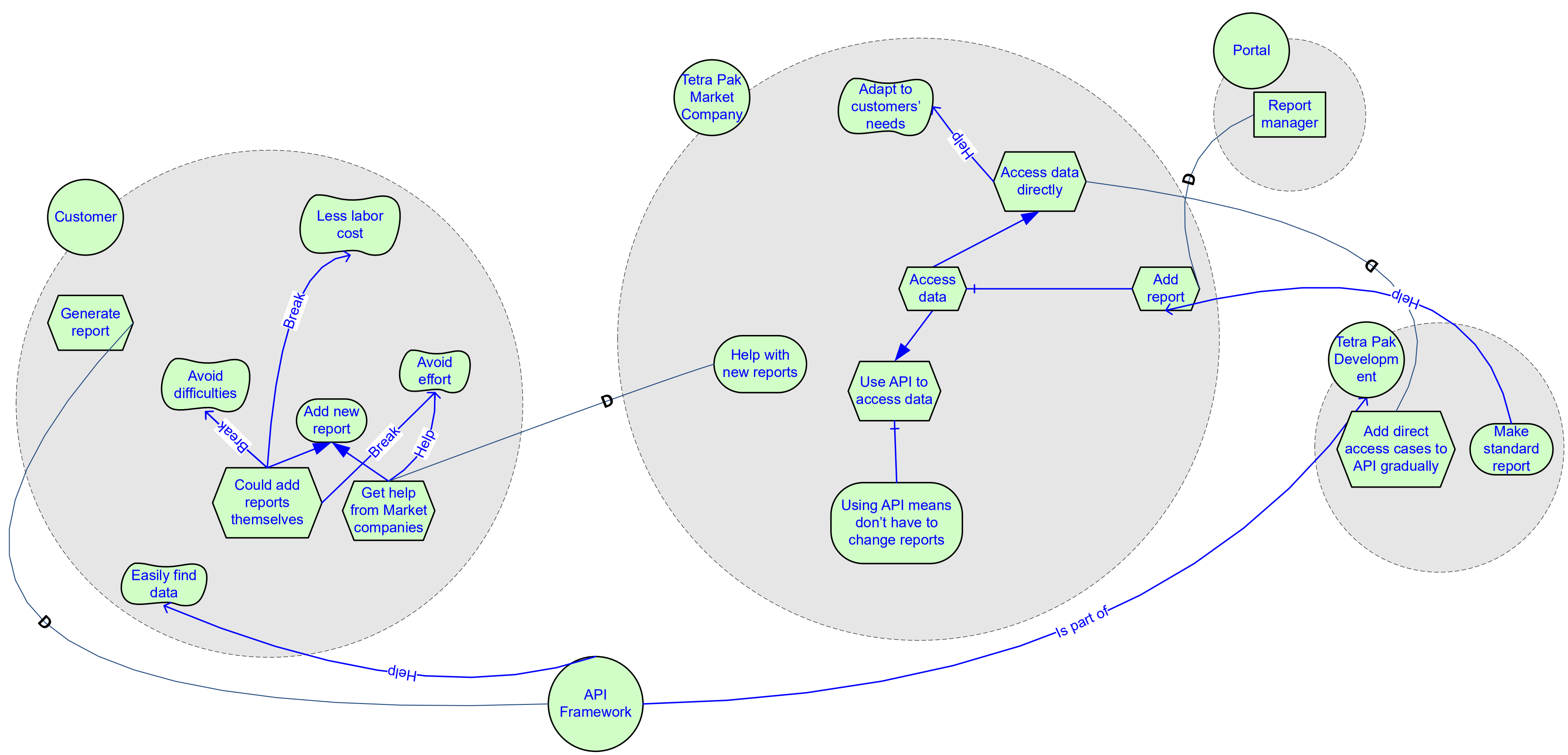

Fig. 23. Application Software layer of Tetra Pak API ecosystem

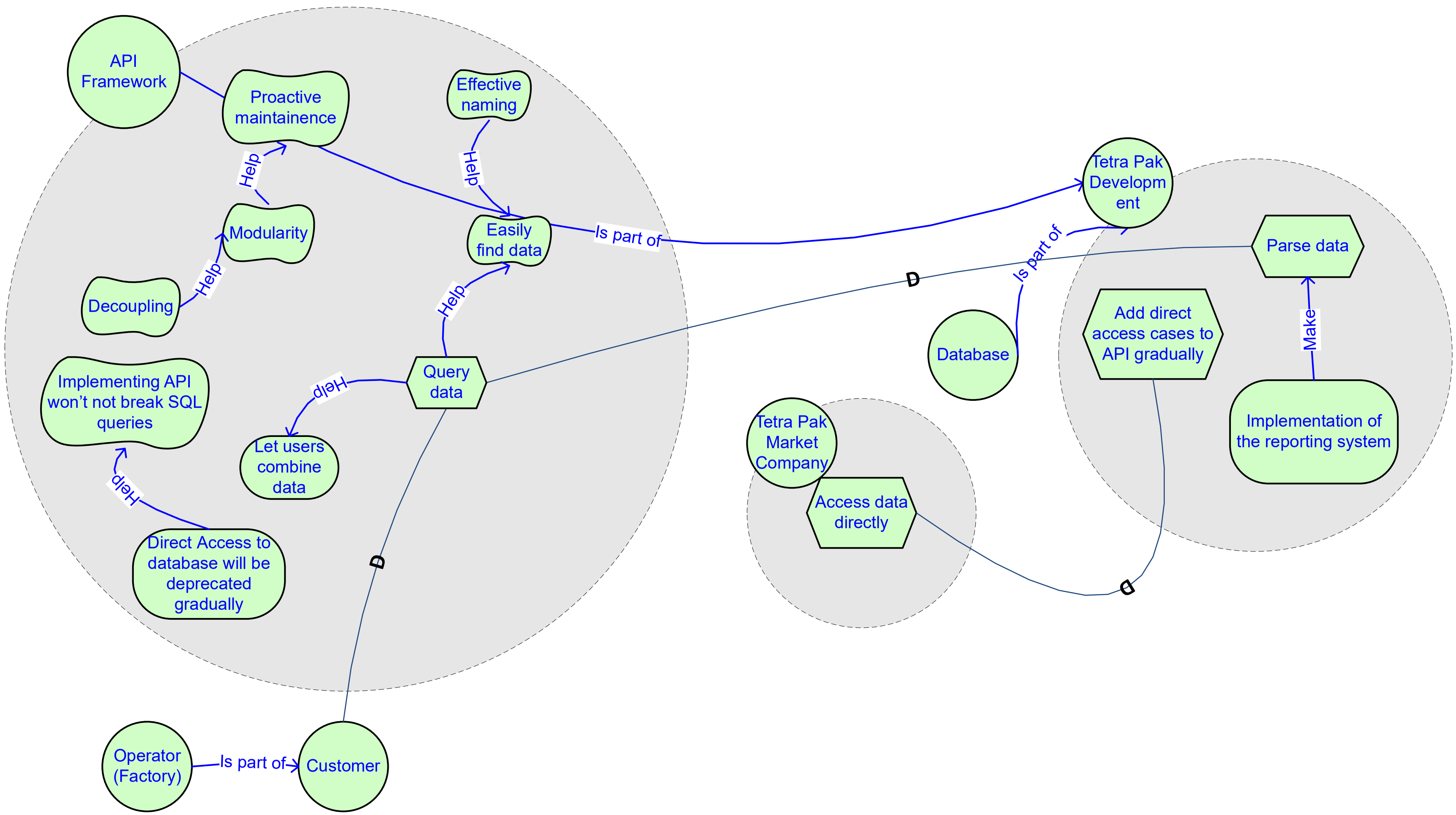

Fig. 24. API layer of Tetra Pak API ecosystem

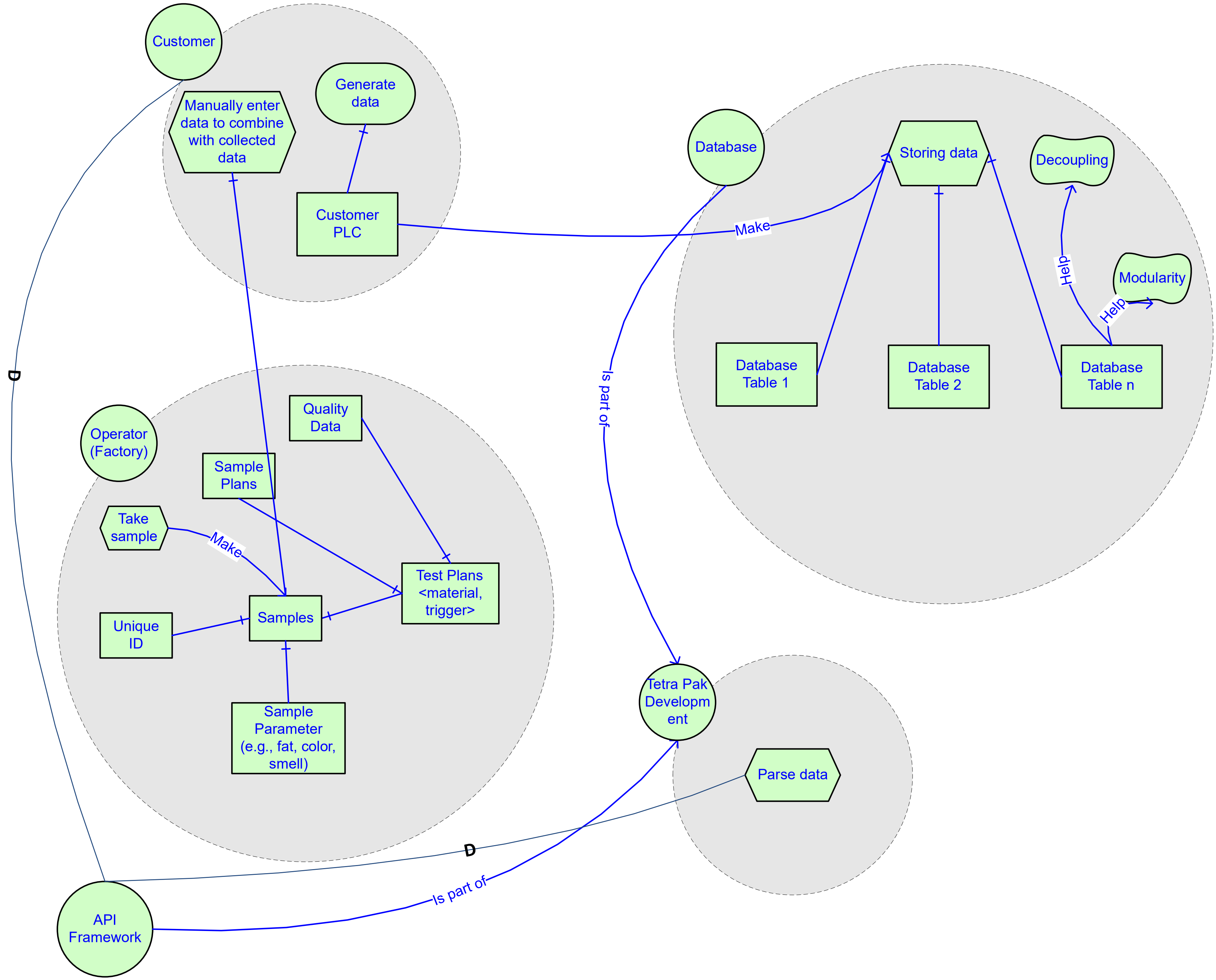

Fig. 25. Business Assets layer of Tetra Pak API ecosystem

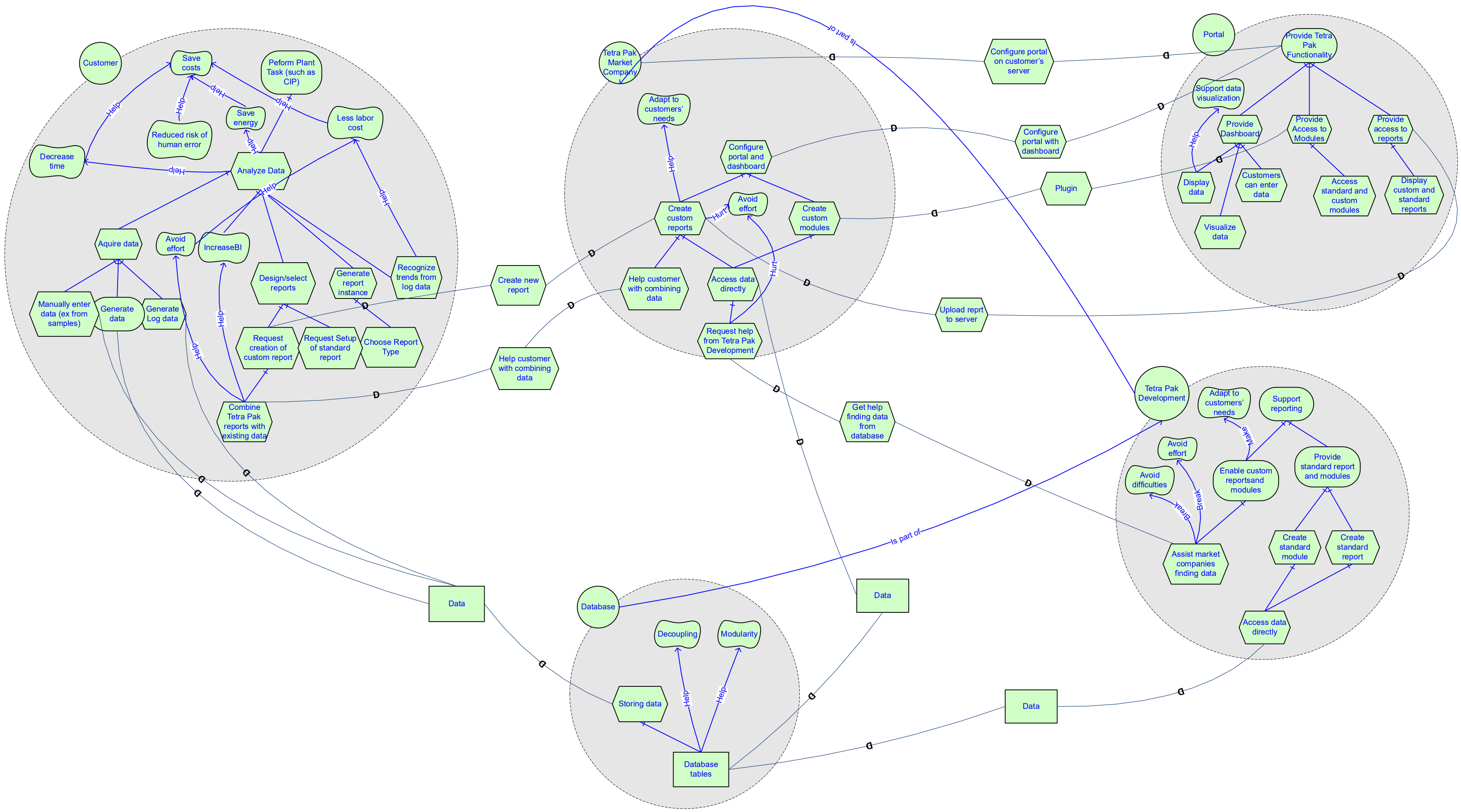

Fig. 26. As-is version of Tetra Pak API ecosystem

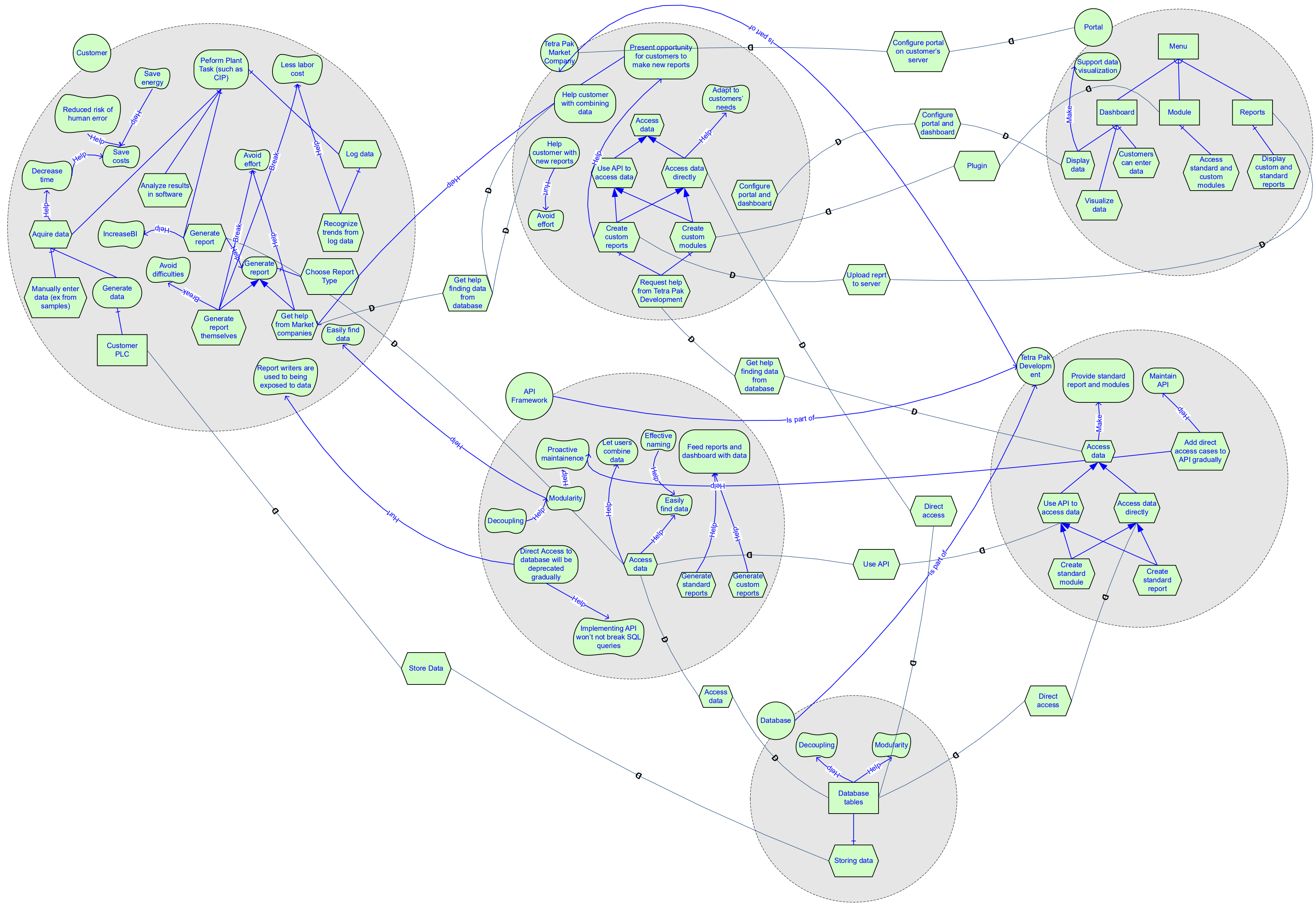

Fig. 27. Near near future version of Tetra Pak API ecosystem

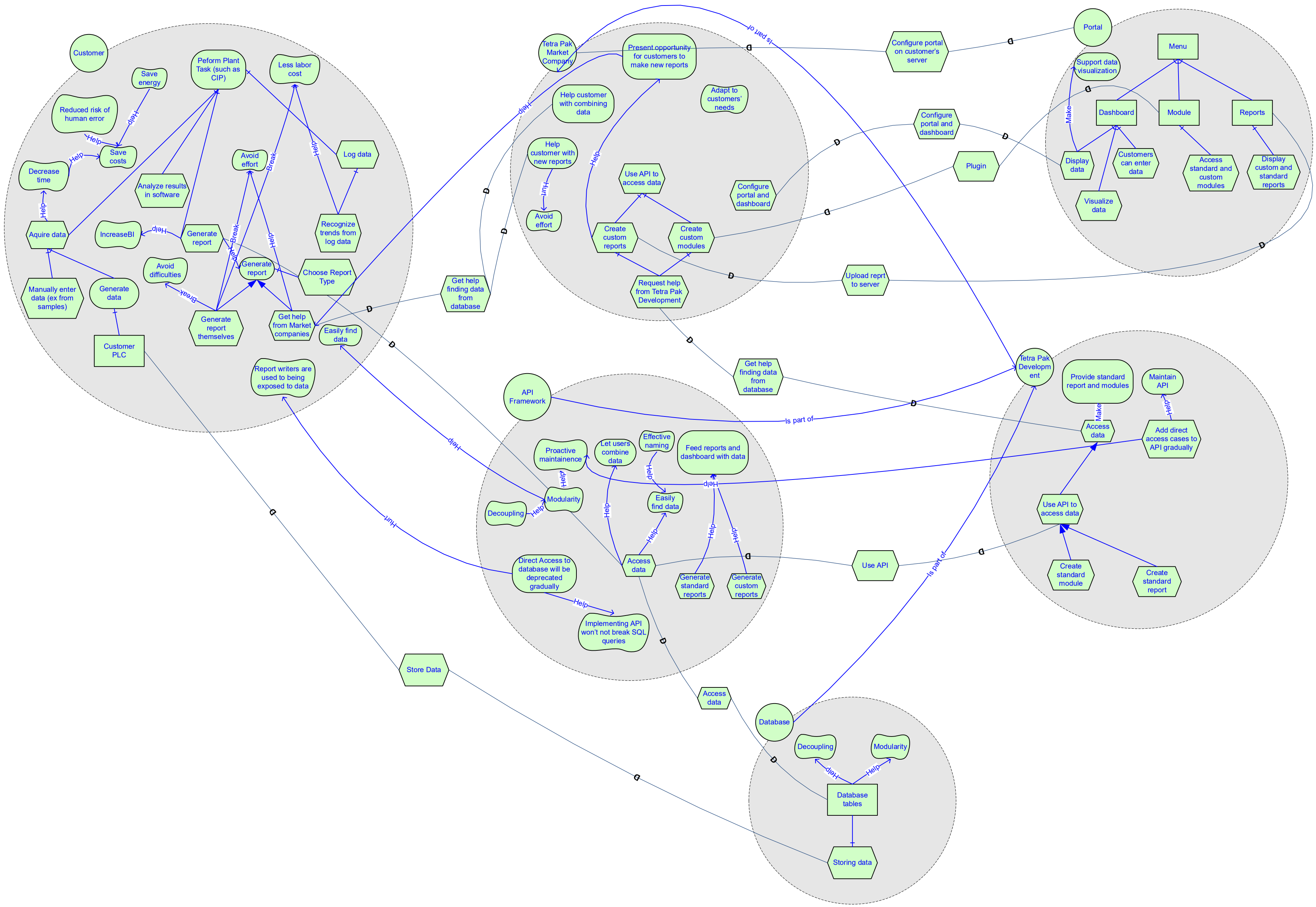

Fig. 28. Near future of Tetra Pak API ecosystem

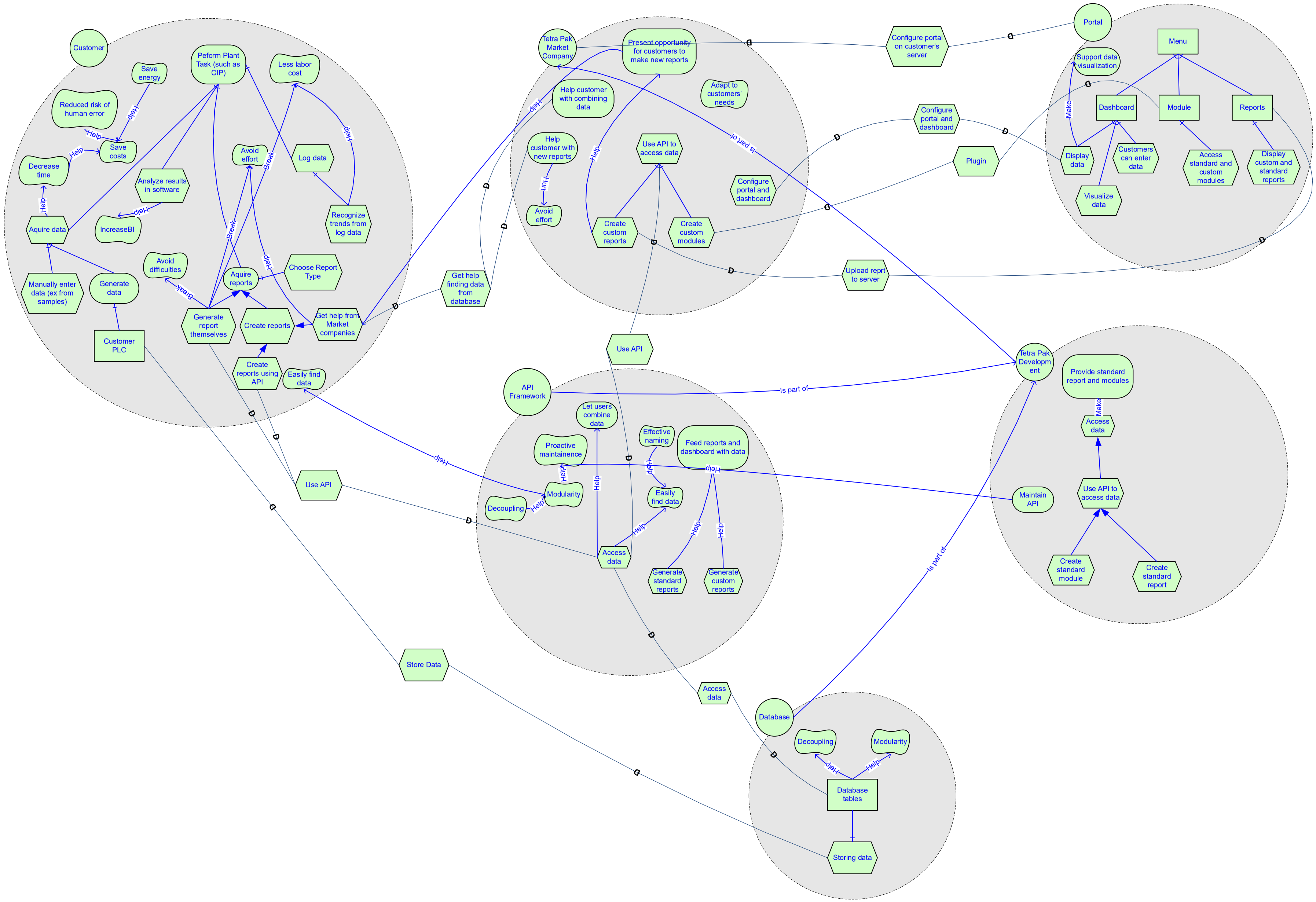

Fig. 29. Distant future version of Tetra Pak API ecosystem

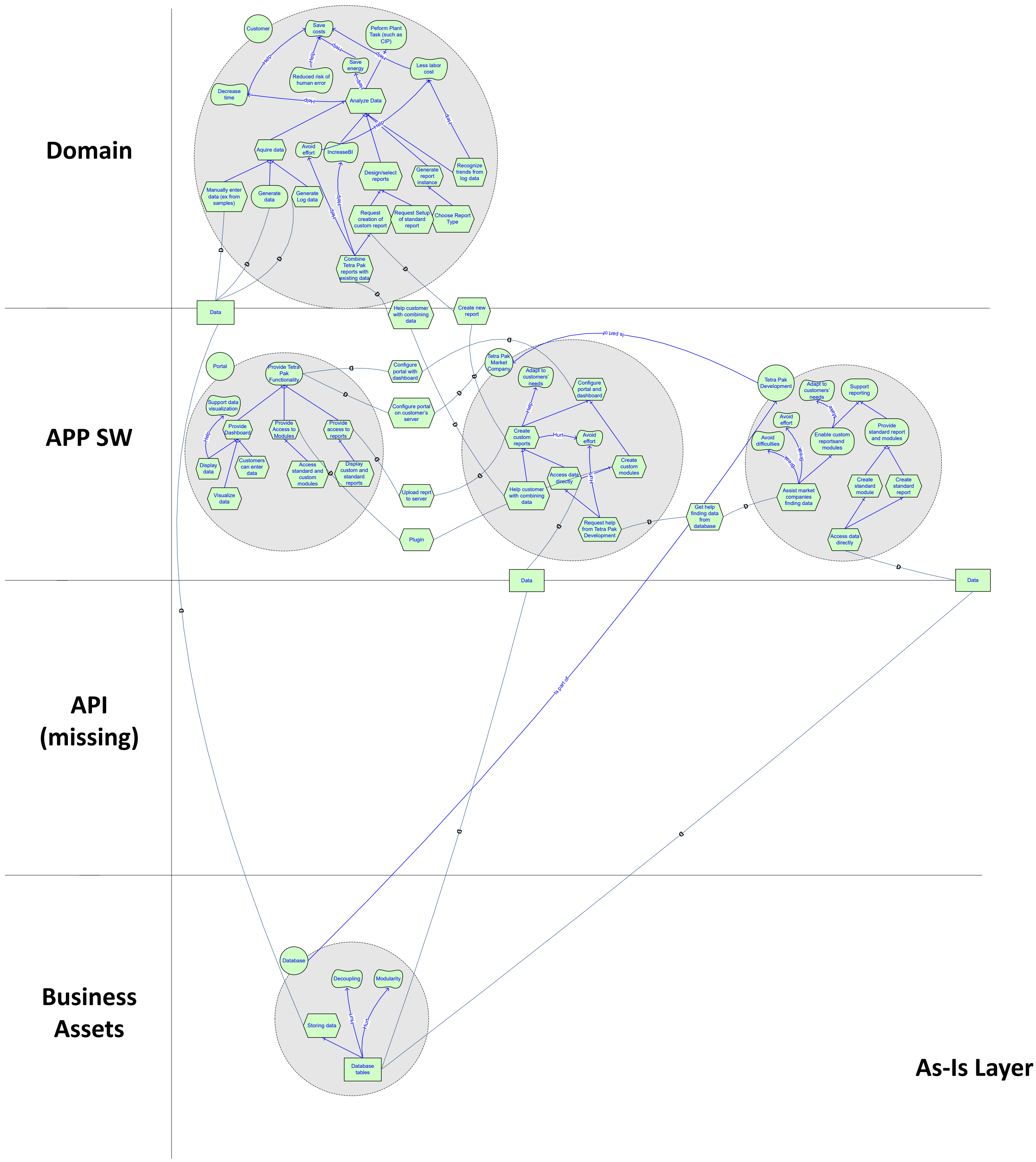

Fig. 30. As-is layered version of Tetra Pak API ecosystem

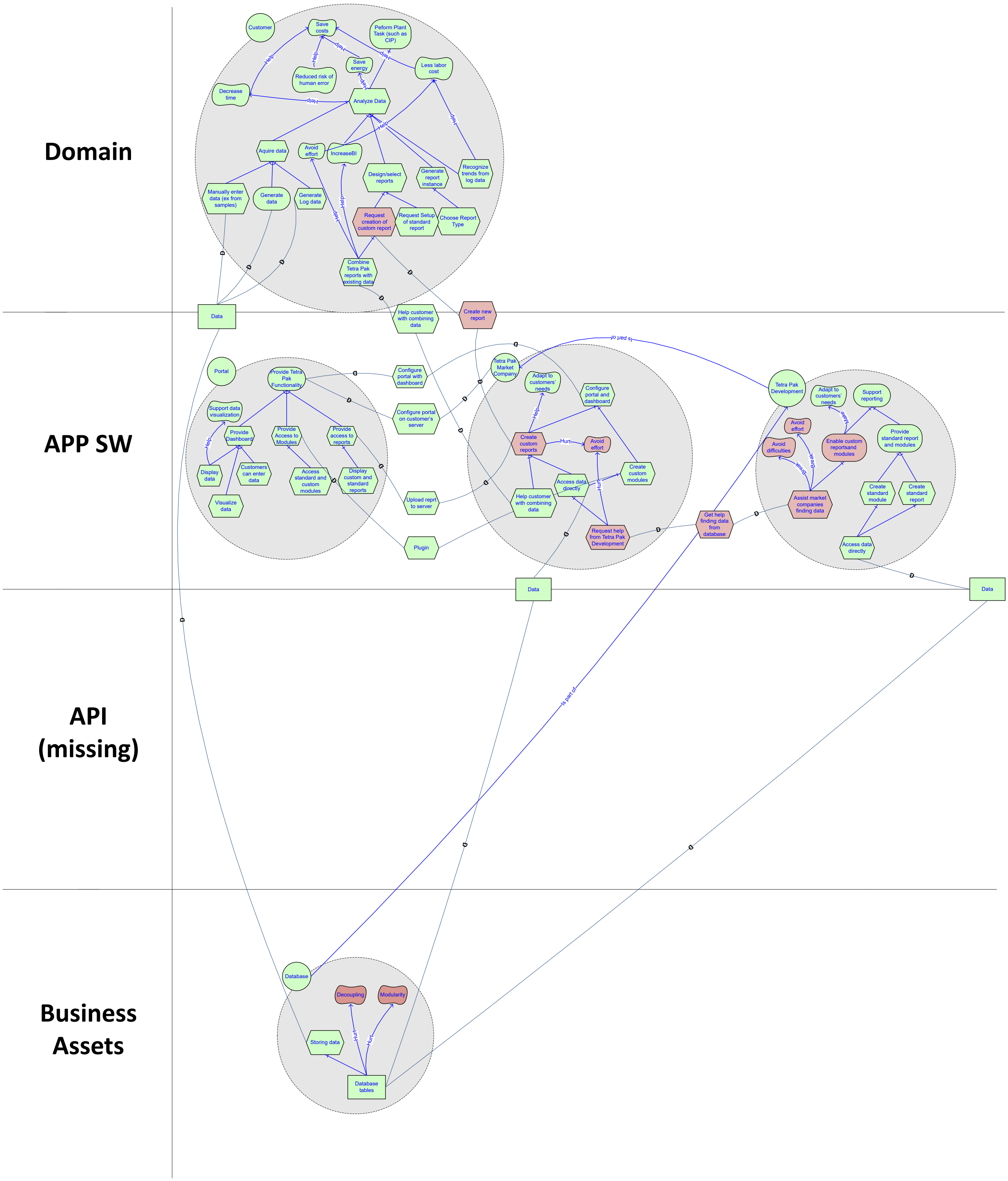

Fig. 31. As-is layered and colored version of Tetra Pak API ecosystem

Fig. 32. Near near future layered and colored version of Tetra Pak API ecosystem

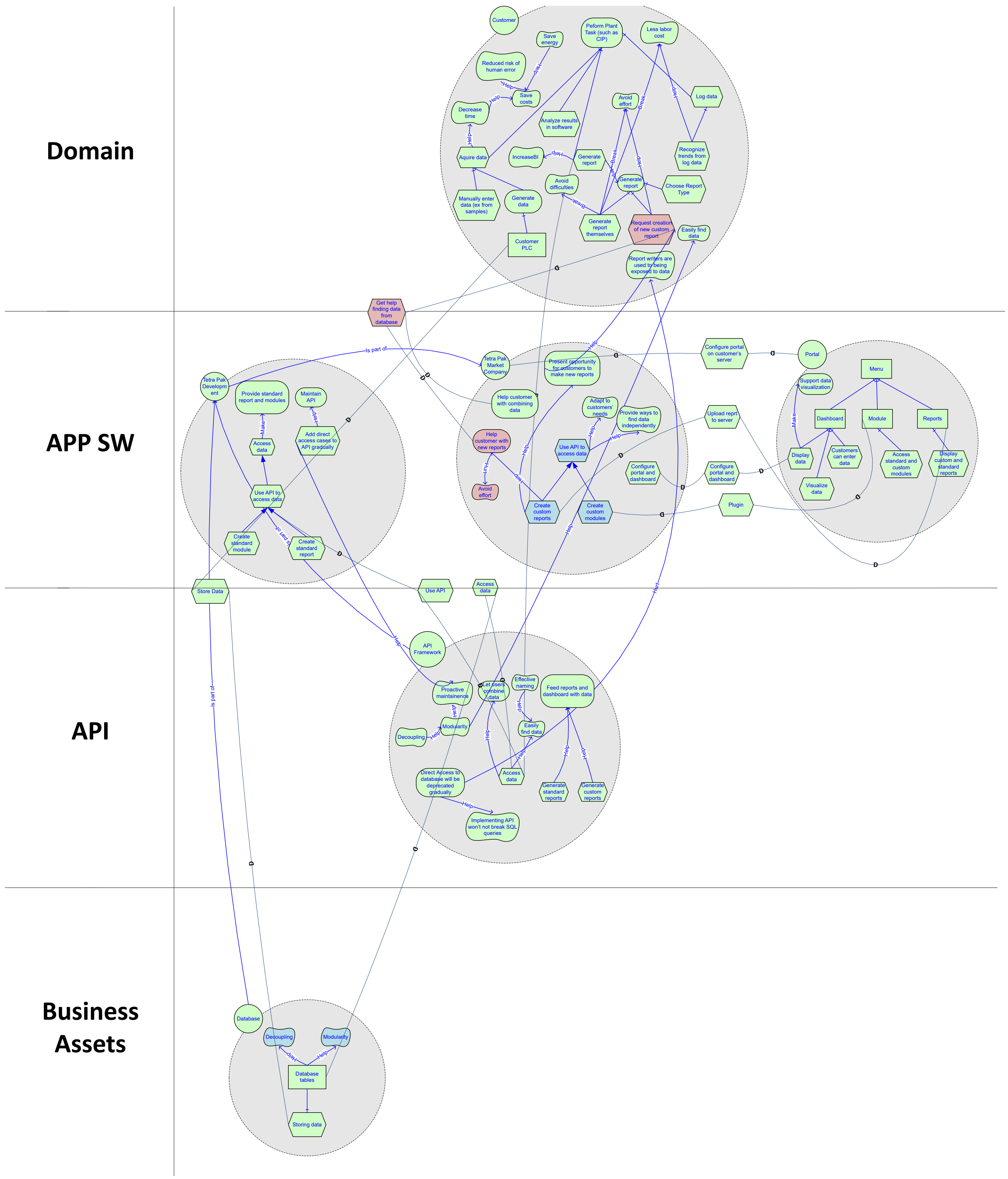

Fig. 33. Near future layered and colored of Tetra Pak API ecosystem

Fig. 34. Distant future layered and colored version of Tetra Pak API ecosystem

2) Axis Communications:

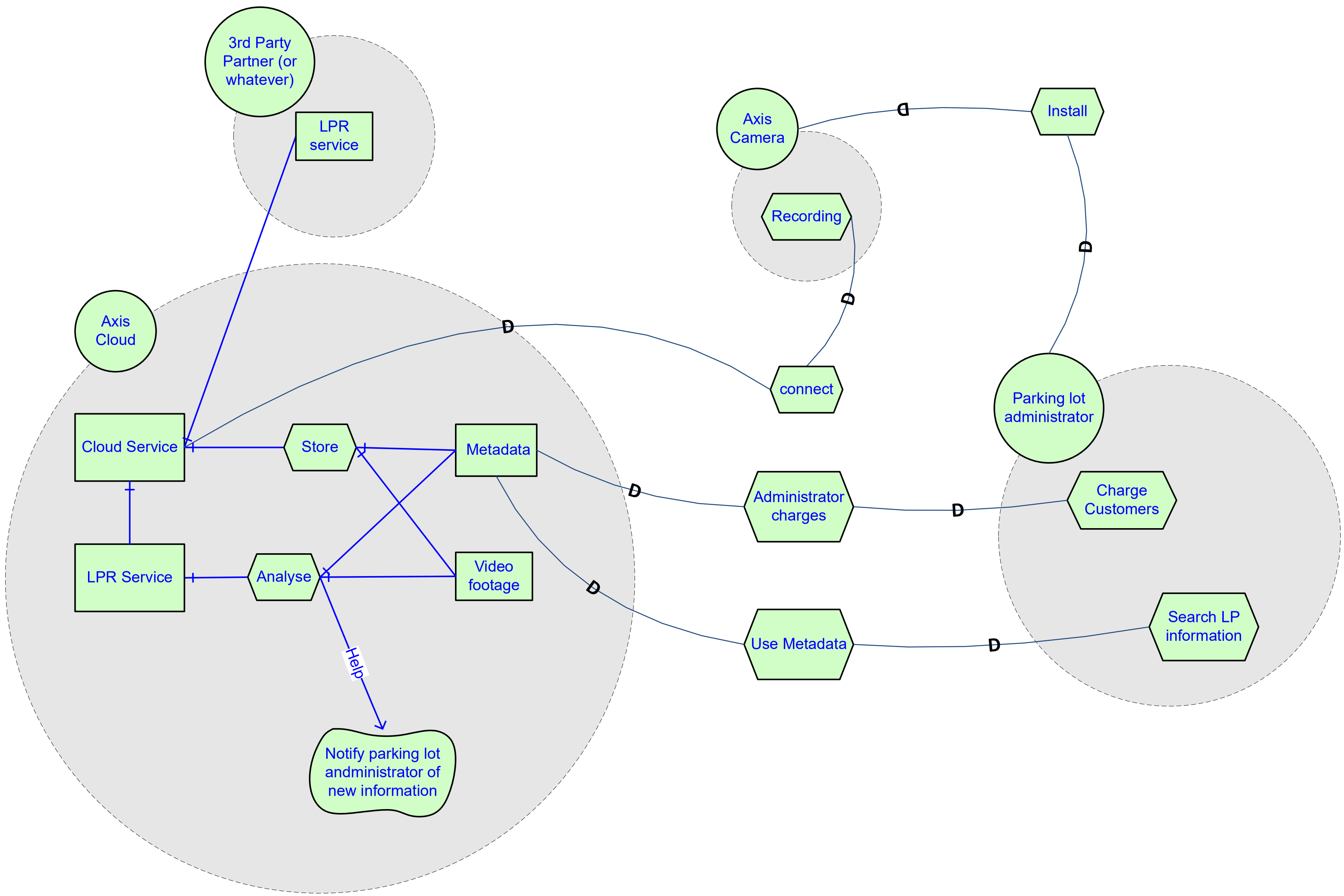

Fig. 35. Version 1 of Axis Use-case

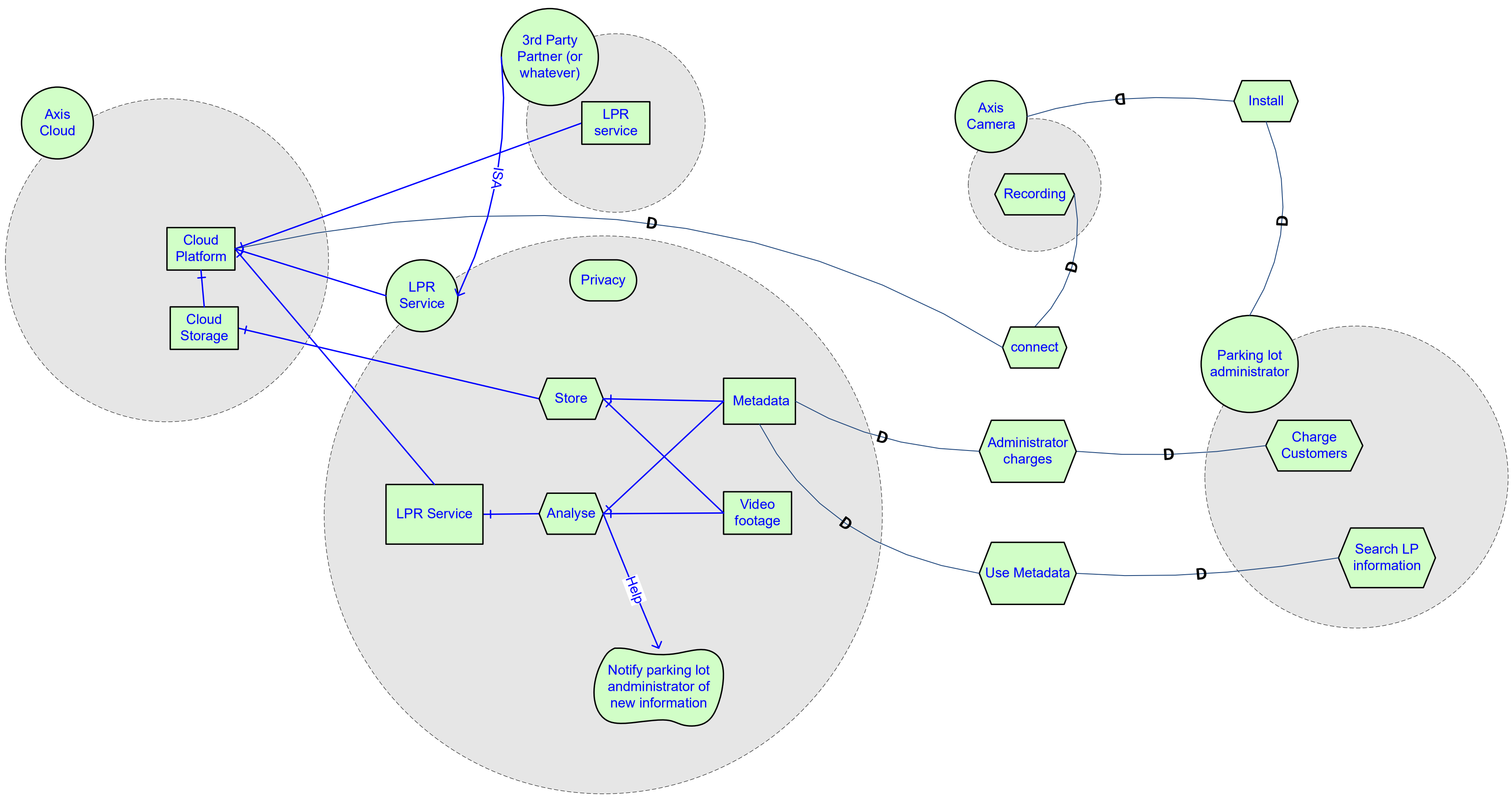

Fig. 36. Version 2 of Axis Use-case

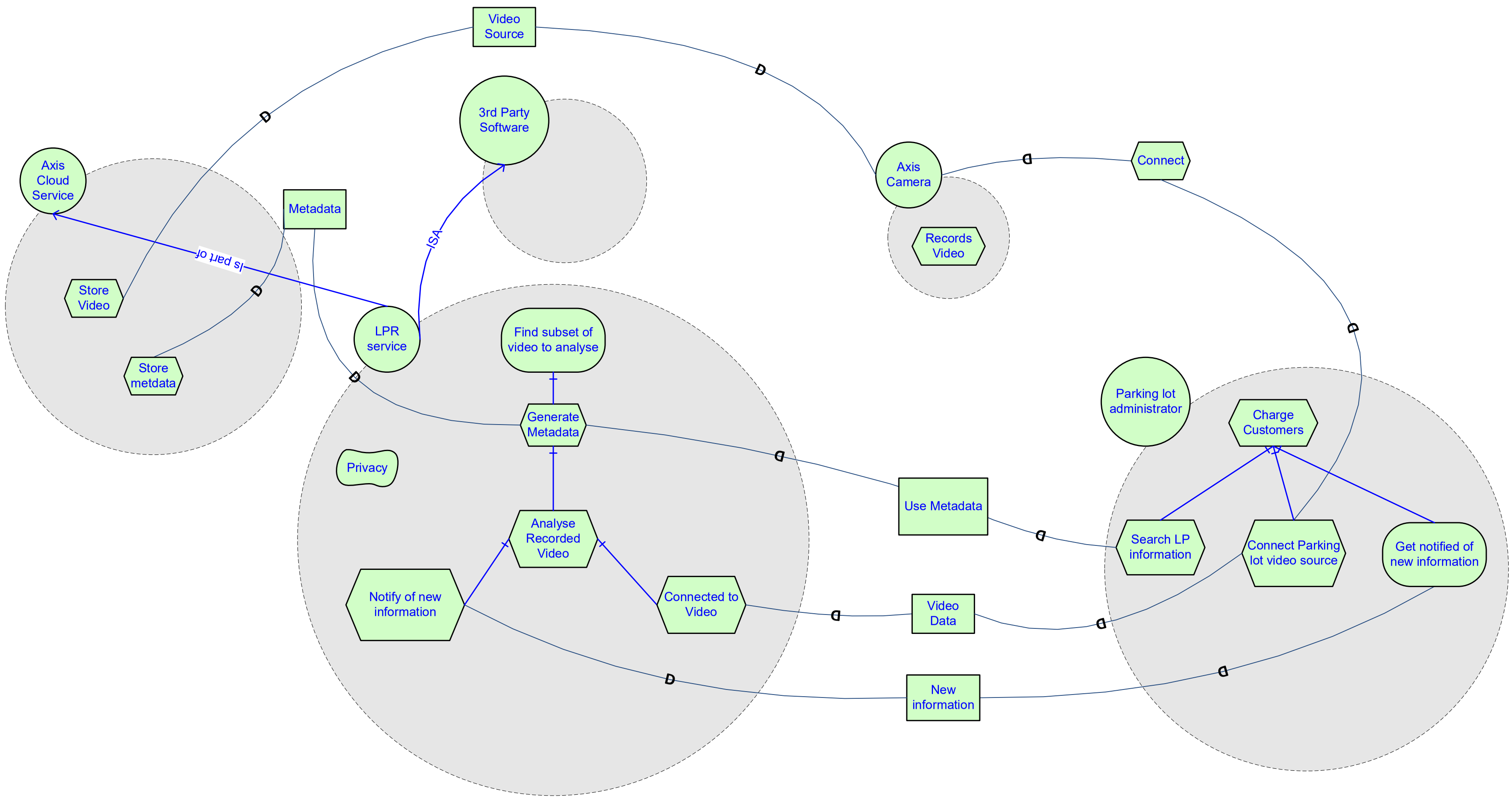

Fig. 37. Version 3 of Axis Use-case

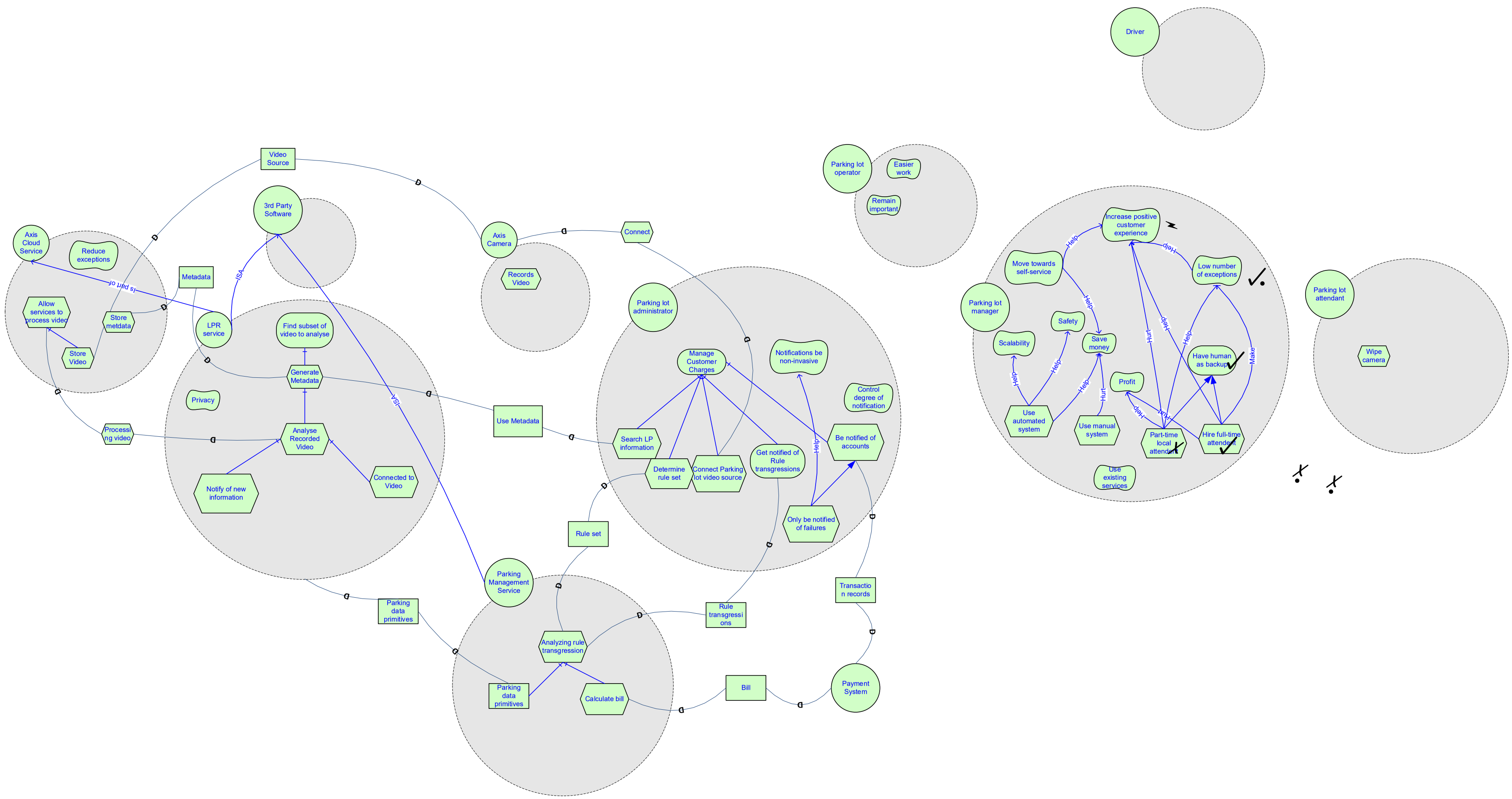

Fig. 38. Workshop Version 4 of Axis Use-Case

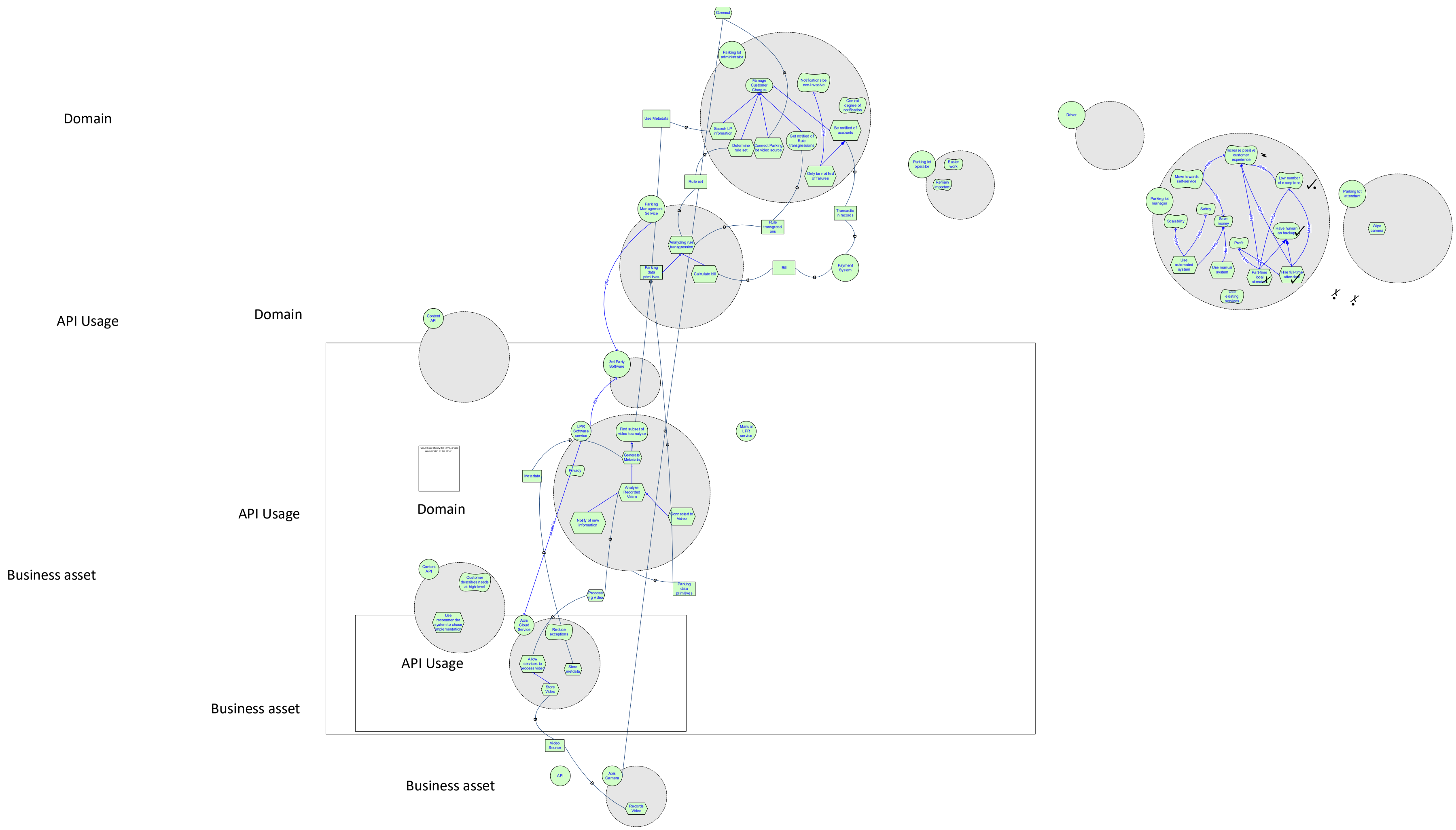

Fig. 39. Workshop Version 5 of Axis Use-Case

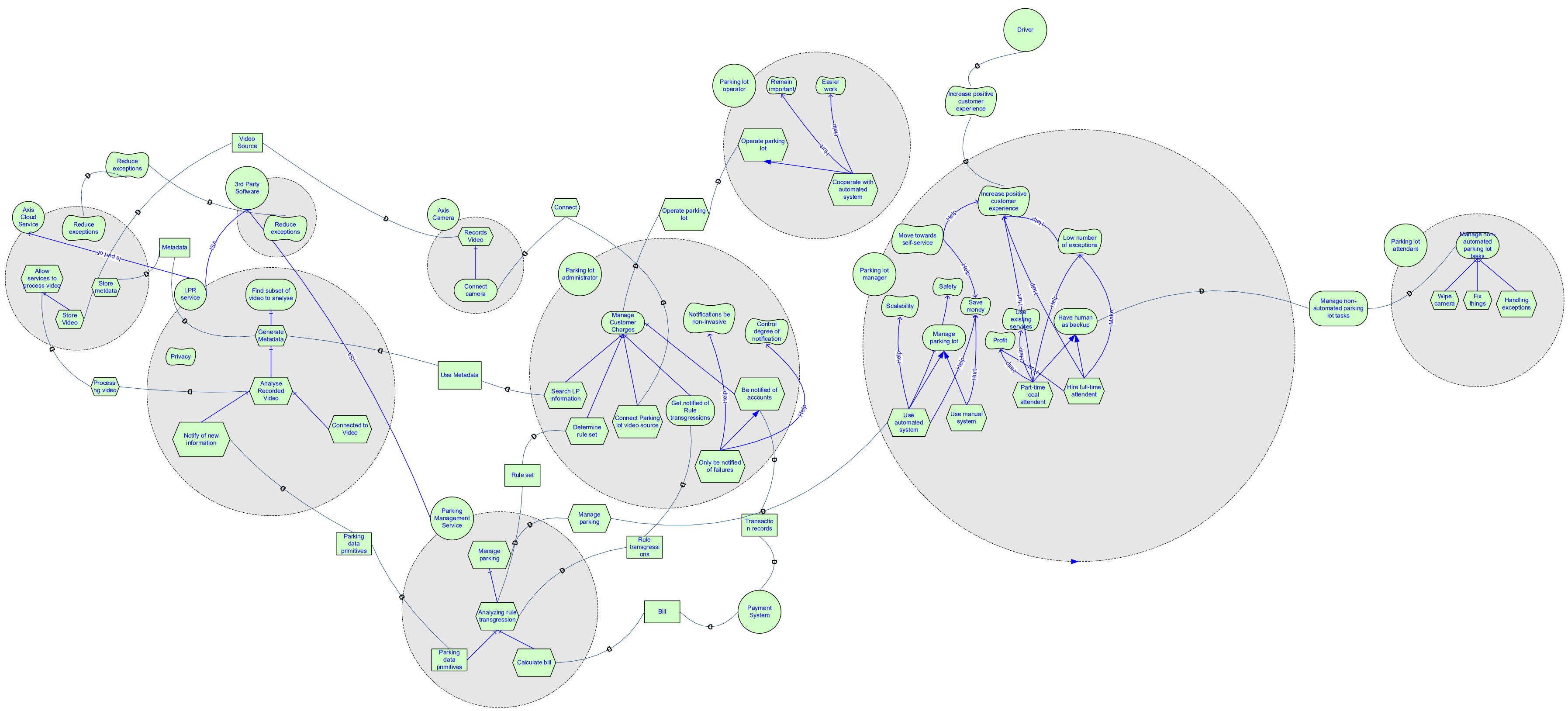

Fig. 40. Version 6 of Axis Use-Case Based On Workshop Feedback

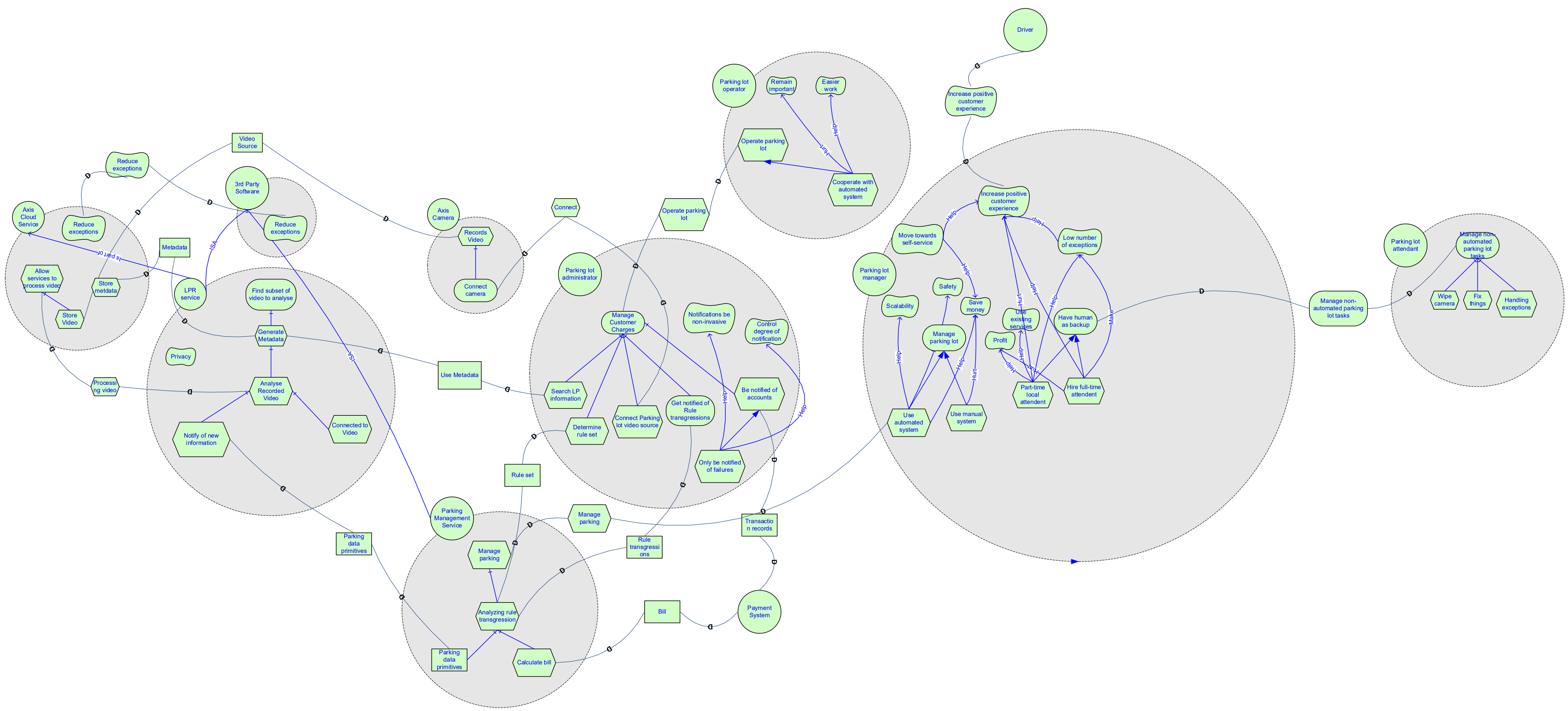

Fig. 41. Version 7 of Axis Use-Case Based On Workshop Feedback

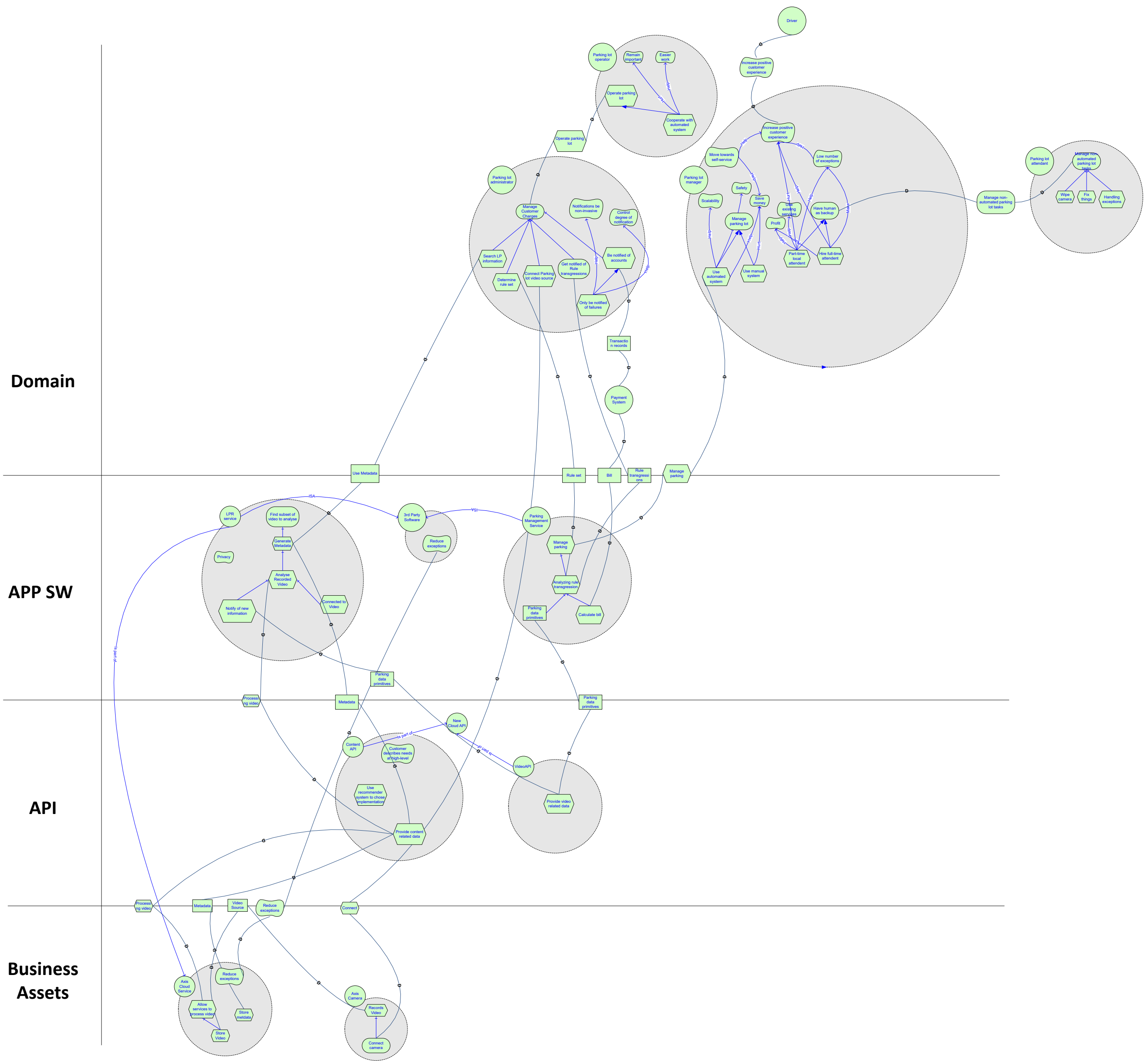

Fig. 42. Initial Layered Version of Axis API ecosystem

Fig. 43. Layered Version of Axis API ecosystem - Pre-existing API perspective

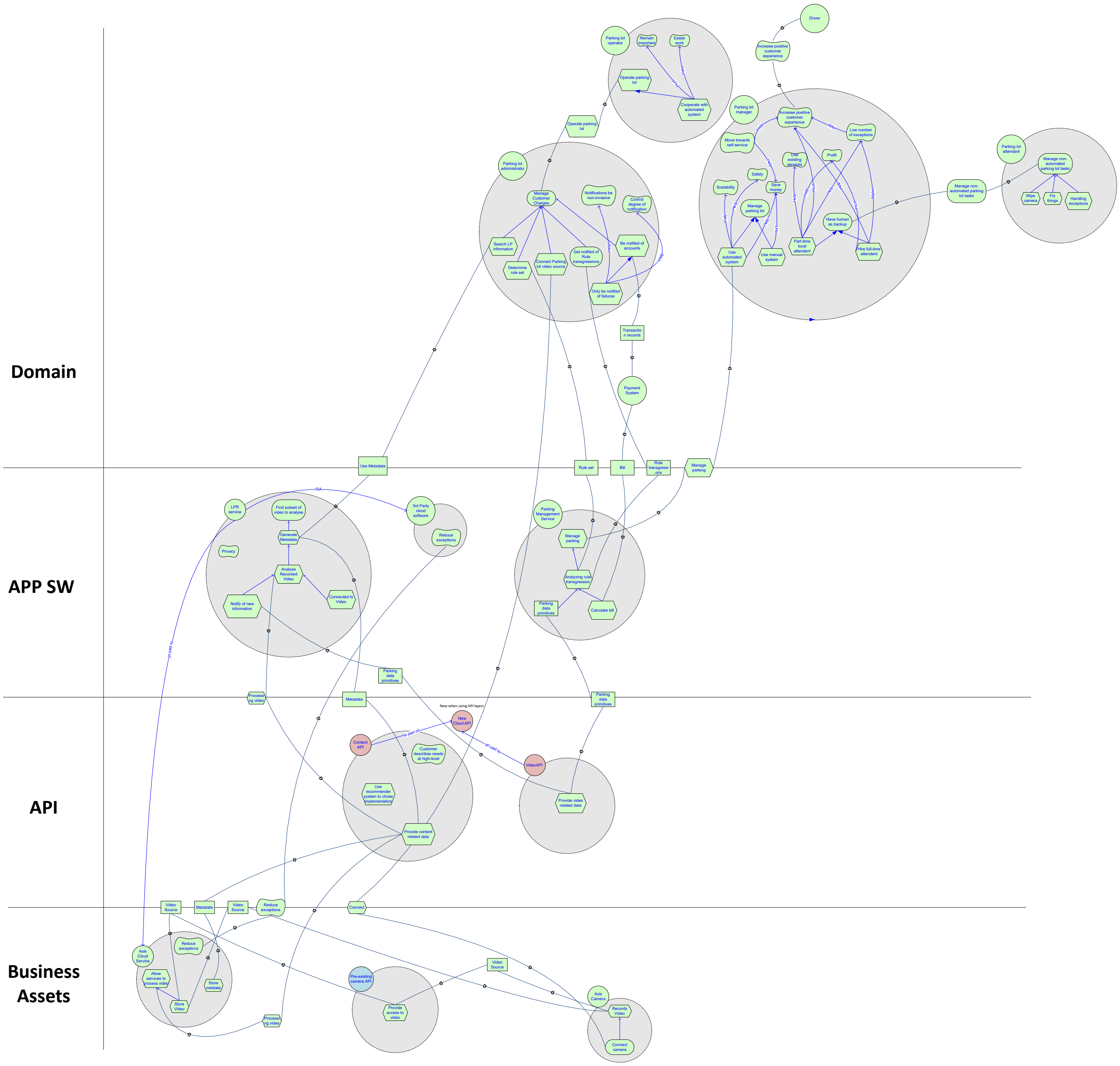

Fig. 44. Layered Version of Axis API ecosystem - Cloud API perspective

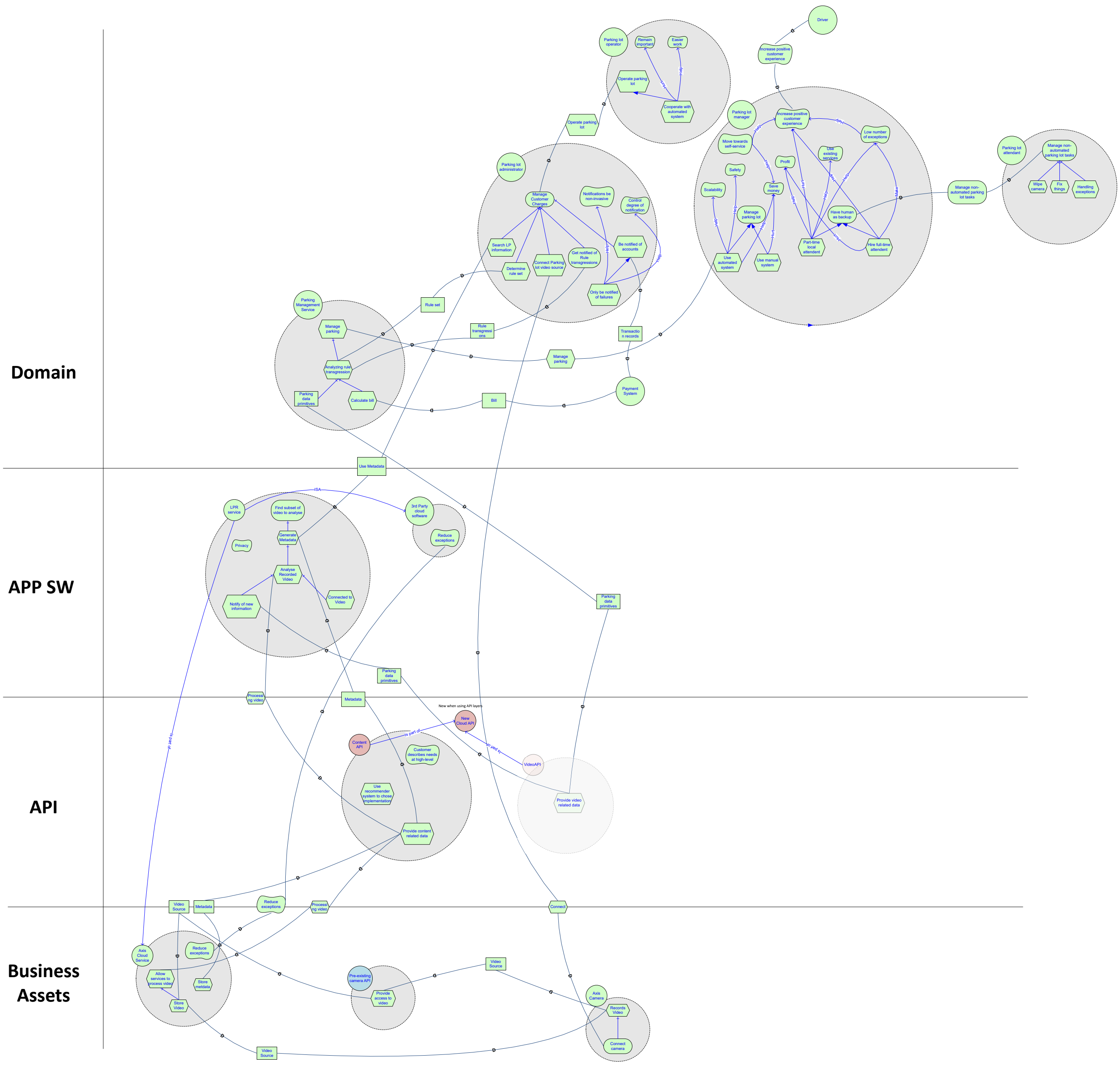

Fig. 45. Layered Version of Axis API ecosystem - Content API perspective

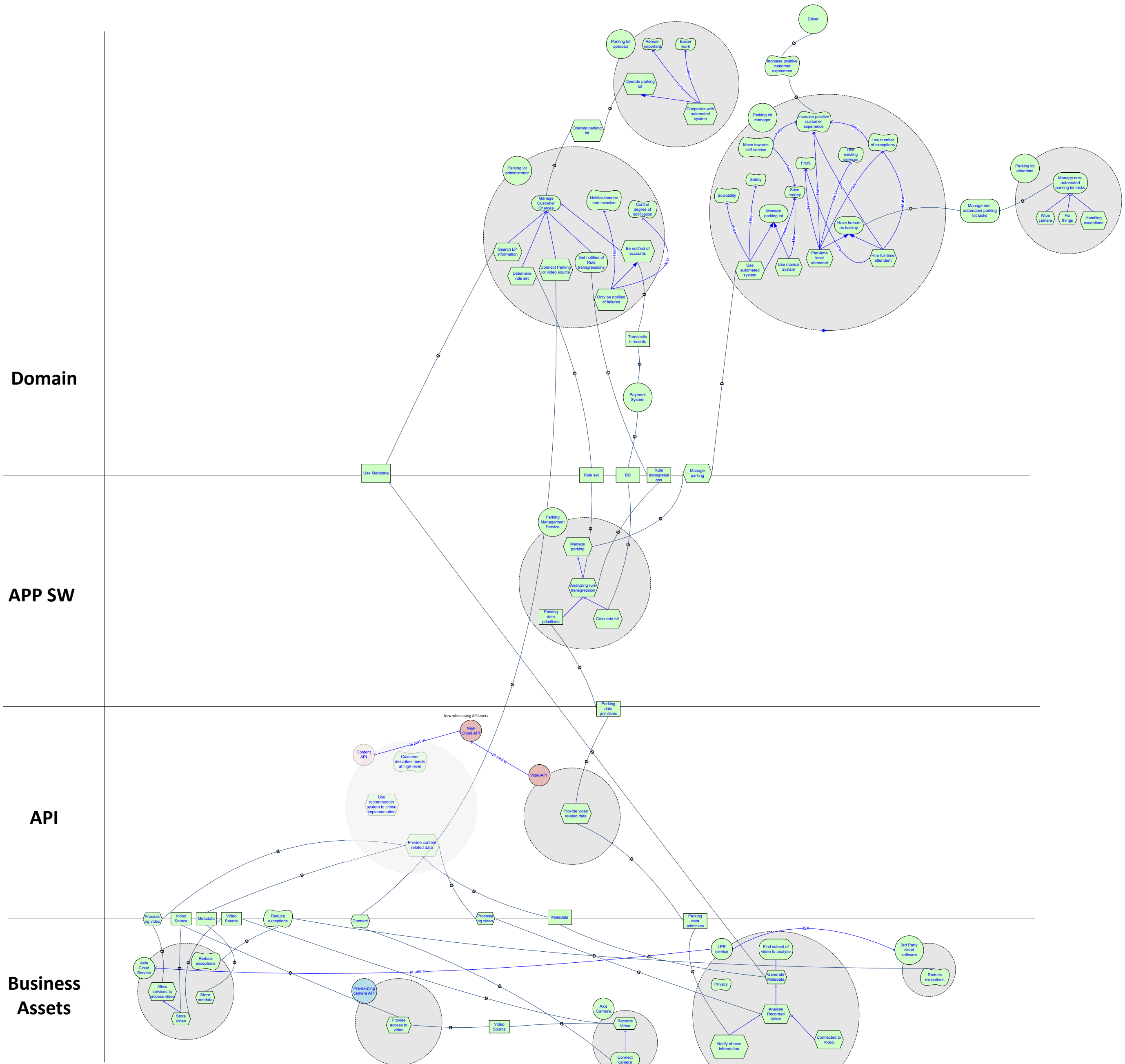

Fig. 46. Layered Version of Axis API ecosystem - Video API perspective

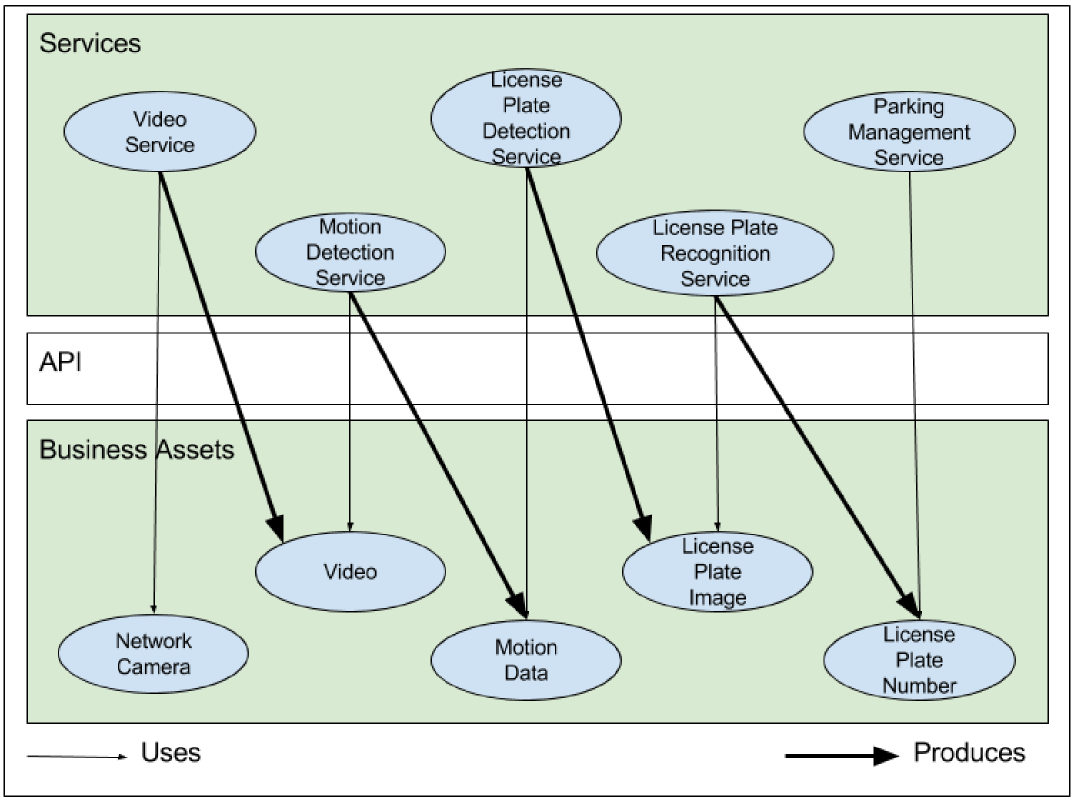

Fig. 47. Focused-view created by Axis' Participant

Acknowledgment

The researchers would like to extend their gratitude towards the supervisors Imed Hammouda, Jennifer Horkoff whom provided assistance with direction of the research as well as helping with modeling API ecosystems. It supported the group as a guidance regarding both the research questions and theories included in the report.

References

[1] W. Zhu, A. Daniel, A. Das, S. Dickerson, J. Falkl, K. Sanders, D. Shetty, and C. Wood, *Exposing and Managing Enterprise Services with IBM API Management*. IBM WebSphere, 2014, ch. 1.

[2] N. Brown, M. Graham, S. Talapatra, and S. Dale, "Social network application programming interface," Jan. 1 2013, uS Patent 8,347,322. [Online]. Available: https://www.google.com/patents/US8347322

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

[4] Oracle, "Making Money Through API Exposure," 2014. [Online]. Available: http://www.oracle.com/us/industries/communications/comm-making-money-wp-1696335.pdf

[5] C. Werner, "Software ecosystem," *Science*, vol. 2, no. 14, pp. 1–26, 2012.

[6] J. Stylos and B. Myers, "Mapping the space of API design decisions," *Proceedings - IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2007*, pp. 50–57, 2007.

[7] J. Joshua, D. Alao, S. Okolie, and O. Awodele, "Software Ecosystem: Features, Benefits and Challenges," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 8, pp. 1–6, 2013. [Online]. Available: http://thesai.org/Publications/ViewPaper?Volume=4{\&}Issue=8{\&}Code=IJACSA{\&}SerialNo=33

[8] E. Yu and S. Deng, "Understanding software ecosystems: A strategic modeling approach," *CEUR Workshop Proceedings*, vol. 746, pp. 65–76, 2011.

[9] S. Jansen, E. Handoyo, and C. Alves, "Scientists' needs in modelling software ecosystems," *ACM International Conference Proceeding Series*, vol. 07-11-Sept, pp. 1–6, 2015. [Online]. Available: http://dx.doi.org/10.1145/2797433.2797479

[10] V. Boucharas, S. Jansen, and S. Brinkkemper, "Formalizing software ecosystem modeling," *Proceedings of the 1st international workshop on Open component ecosystems IWOCE 09*, vol. 19, no. 2, p. 41, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1595800.1595807

[11] M. H. Sadi and E. Yu, "Designing software ecosystems: How can modeling techniques help?" *Lecture Notes in Business Information Processing*, vol. 214, pp. 360–375, 2015. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84937425596{\&}partnerID=40{\&}md5=f479c7666a4de08405afed85480d63c0

[12] V. Boucharas, S. Jansen, and S. Brinkkemper, "Formalizing software ecosystem modeling," *Proceedings of the 1st international workshop on Open component ecosystems IWOCE 09*, vol. 19, no. 2, p. 41, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1595800.1595807

[13] A. Osterwalder, C. Parent, and Y. Pigneur, "Setting up an ontology of business models," *CEUR Workshop Proceedings*, vol. 125, 2004.

[14] V. Allee, "Value network analysis and value conversion of tangible and intangible assets," *Journal of Intellectual Capital*, vol. 9, no. 1, pp. 5–24, 2008. [Online]. Available: http://dx.doi.org/10.1108/14691930810845777

[15] W. D. Jacobson D., Brail G. O'Reilly Media, 2011, p. 150.

[16] K. Manikas and K. M. Hansen, "Software ecosystems-A systematic literature review," *Journal of Systems and Software*, vol. 86, no. 5, pp. 1294–1306, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2012.12.026

[17] "Emerging perspectives to api strategy," contact for reference.

[18] E. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," *Proceedings of ISRE '97: 3rd IEEE International Symposium on Requirements Engineering*, pp. 226–235, 1997.

[19] J. Horkoff, N. Maiden, and J. Lockerbie, "Creativity and Goal Modeling for Software Requirements Engineering," *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, pp. 165–168, 2015. [Online]. Available: http://dx.doi.org/10.1145/2757226.2764544

[20] J. Noppen, J. Noppen, P. van Den Broek, P. van Den Broek, M. Aksit, and M. Aksit, "Requirements Engineering: Foundation for Software Quality," *Requirements Engineering: Foundation for Software Quality*, vol. 4542, no. January, pp. 247 – 261, 2007. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-540-73031-6

[21] N. R. Darwish and B. S. M. Zohdy, "Goal Modeling Techniques for Requirements Engineering," vol. 14, no. 7, pp. 739–747, 2016.

[22] V. Chiprianov, Y. Kermarrec, S. Rouvrais, and J. Simonin, "Extending enterprise architecture modeling languages for domain specificity and collaboration: Application to telecommunication service design," *Softw. Syst. Model.*, vol. 13, no. 3, pp. 963–974, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1007/s10270-012-0298-0

[23] L. Guide, "iStar 2.0 Language Guide," pp. 1–15.

[24] M. H. Sadi and E. Yu, "Analyzing the evolution of software development: From creative chaos to software ecosystems," in *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, May 2014, pp. 1–11.

[25] C. B. Seasman, *Qualitative Methods in Empirical Studies of Software Engineering*. Springer, July/August 1999, vol. 25.

[26] J. Kitzinger, "Qualitative research: Introducing focus groups," *BMJ*, vol. 311, no. 7000, pp. 299–302, 1995. [Online]. Available: http://www.bmj.com/content/311/7000/299

[27] H. T. Lawless and H. Heymann, *Qualitative Consumer Research Methods*. New York, NY: Springer New York, 2010, pp. 379–405. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-6488-5_16

[28] J. Frontiera, "Leadership and organizational culture transformation in professional sport," *Journal of Leadership & Organizational Studies*, vol. 17, no. 1, pp. 71–86, 2010. [Online]. Available: http://dx.doi.org/10.1177/1548051809345253