Management and Governance of External Developer Platforms – at the example of Akamai, Inc. and Uber Technologies, Inc.

By

Christian Umbach

Submitted to the System Design & Management Program
In Partial Fulfilment of the Requirements for the Degree of

Master of Science in Engineering & Management

at the

Massachusetts Institute of Technology

February 2017

© 2017 Christian Umbach. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly, paper and electronic copies of this thesis document, in whole or in part, in any medium now known or hereafter created.

Written by

Signature redacted

Christian Umbach
MIT System Design & Management Program
January 23, 2017

Certified by

Signature redacted

Jeanne Ross
Thesis Advisor
MIT Sloan School of Management

Accepted by

Signature redacted

Warren Seering
Weber-Shaughness Professor of Mechanical Engineering
Management and Governance of External Developer Platforms – at the example of Akamai, Inc. and Uber Technologies, Inc.

By

Christian Umbach

Submitted to the System Design & Management Program on January 23, 2017 in partial fulfilment of the requirements for the Degree of Master of Science in Engineering & Management at the Massachusetts Institute of Technology.

ABSTRACT

How do companies manage and govern external developer platforms? Drawing on platform ecosystem, product management, and IT governance theory a comparison of the developer platform approaches of two technology firms, Akamai, Inc. and Uber Technologies, Inc. is presented. Internal organizational structures and processes are evaluated in combination with external governance mechanism to develop and manage third party partner relations.

The case studies test existing IT governance and app developer platform theory with regards to its applicability to platforms for developer products such as APIs. The two teams are analyzed along the dimensions of organizational structure, internal governance process and risk management, external governance and partner relations, as well as KPIs for team steering. A set of key governance mechanisms is identified as success factors for limiting risks around open developer platforms: Internally, API standards and working groups supported by a high degree of interconnectedness support governance and avoid de-acceleration around a company’s product development process. Externally, legal agreements around terms of use shift control from the developer to the platform provider. Access limitations can serve as a supporting gatekeeper for access, or punitive penalty in case of violations of agreements. In support of contractual agreements, transparent developer communication around a firm’s intended fields of innovation help avoid discords between developer platform providers and its users. Results are discussed.

Thesis Supervisor: Jeanne Ross

Title: Director of MIT Center for Information Systems Research and Principal Research Scientist
Bibliographical Note

Mr. Umbach is leading product for XapiX, Inc. which he co-founded in 2015. The company builds a developer tool facilitating API integration. Previously, he was a founding member of Lufthansa Innovation Hub GmbH where he led the OpenAPI development as a product manager. He started his career at Lufthansa Group in 2009 where he initially held assignments related to managing internal projects for the CEO & CFO of Lufthansa Cargo.

As a Graduate Fellow in System Design & Management at Massachusetts Institute of Technology, Mr. Umbach led a team dedicated to digital technologies as a Senior Engineering Specialist for Hyperloop Transportation Technologies. During the summer break at MIT, he joined Uber Technologies, Inc. in San Francisco, CA as a Developer Advocate on the Developer Platform team.

Mr. Umbach received a Bachelor of Science degree in Aviation Management from the European Business School, Oestrich-Winkel, Germany in June 2012.
Acknowledgement

I am truly thankful for the opportunity to work with some of the smartest people in the world, many of whom I can also count among my personal friends. Without the support of my family, best friends, mentors, business partners, fellow students, and professors I would not have dared to take this level of risk in my life – something that has led me to MIT and is rewarding me beyond.

As they inspire me to go beyond the ordinary, I am dedicated to pass on this inspiration to those who are excited to equally build new, impactful contributions for their communities, countries, and the globe.
Contents

Bibliographical Note .............................................................................................................. 3
Acknowledgement ................................................................................................................ 4
Chapter 1: Introduction ............................................................................................................ 8
Chapter 2: Literature Review ................................................................................................ 11
  Underlying Technologies of Developer Platforms ......................................................... 11
  Platform Ecosystems ....................................................................................................... 13
  Product Development Process ...................................................................................... 15
  Governance ..................................................................................................................... 18
Chapter 3: Case Study Methodology .................................................................................. 23
  Company Introduction .................................................................................................... 26
  General Introduction ....................................................................................................... 27
  Organizational Structure ............................................................................................... 27
  Strategic Product Development Evaluation Process .................................................... 27
  Internal Governance Process & Risk Management ....................................................... 28
  External Governance & Partner Relations .................................................................... 28
  Team Steering / KPIs ...................................................................................................... 29
Chapter 4: Case Studies ........................................................................................................ 29
  Chapter 4.1: Akamai, Inc. ............................................................................................... 29
    Company Introduction .................................................................................................... 29
    Introduction of the Open Platform Team .................................................................... 30
    Organizational Structure ............................................................................................ 31
    Strategic Product Development Evaluation Process ................................................ 32
    Internal Governance Process & Risk Management .................................................... 34
    External Governance & Partner Relations ................................................................ 39
    Team Steering / KPIs .................................................................................................... 41
  Chapter 4.2: Uber Technologies, Inc. ............................................................................ 42
    Company Introduction .................................................................................................... 42
    Introduction of Developer Platform Team .................................................................. 43
    Organizational Structure ............................................................................................ 45
    Strategic Product Development Evaluation Process ................................................ 46
List of Figures and Tables

Figure 1: XapiX (2017), Lean API Program, retrieved from www.xapix.io. Used with permission. ................................................................. 16


Figure 3: Akamai (2016), API Best Practices v2.0. Internal Document (excerpt). Used with permission. ........................................................................................................................................... 39

Table 1: Company Comparison for Selected Dimensions of Developer Platform Governance.................................................................................................................................................. 53
Chapter 1: Introduction

API Growth

In recent years, Application Programming Interfaces (APIs) have been at the forefront of enabling technologies that have made it possible for small startups to break up the core services of established firms and thereby challenge their existing business models. Small companies have leveraged this trend by building businesses around specific service components such as payment delivery, location, and distribution of content-specific data such as weather data. Meanwhile, mature technology firms are leveraging APIs and related developer products to build platforms around their core services, which developers leverage as a foundation to build new applications and services on top. Salesforce.com and eBay were among the first technology companies to launch APIs as part of developer programs, encouraging developers to connect to their services and building their own ecosystems (Lane, n.d.). Amazon followed this lead to become an API-first company around 2002 when Jeff Bezos' "Big Mandate" required all internal teams to expose their data and functionality through service interfaces. Those services were designed with the capability to be externalized, and no alternative forms of data flows between internal departments through direct data sharing between teams were permitted (Rowan, 2011).

Programmeableweb.com, a public API directory, listed more than 15,800 APIs in October 2016 after starting to count its first API in 2005 and reaching 5018 in January 2012 and 10,302 in October 2013. Producthunt.com, a more recent source and directory of new product launches, showed more than 17,100 search results in
December 2016 for “API”. Meanwhile, technology firms have increasingly invested in developer platform teams, with examples including dev.twitter.com, developers.google.com, developer.salesforce.com, and developer.amazon.com. The ability to attract developers is a core component in the strategy of technology firms to scale the reach of their core products and thereby achieve growth. Facebook’s Messenger and Slack platforms enable developers to build bots, Amazon’s Alexa allows developers to build skills for the home device, and Uber offers physical services around rides and deliveries to the applications that developers build.

Beyond the core pillars of an API strategy that encompass distributing and promoting a product or leveraging the API as a product itself, platform companies are leveraging APIs to nurture the interactions of stakeholders within their network. Similarly, APIs can be leveraged to further strengthen a company’s core product as in the case of map providers such as Mapbox and Google Maps. Here, data around usage flows back into the core product to include insights from usage into a more comprehensive offering. Furthermore, APIs can help distribute and nurture the product, such as by leveraging social login functionality of networks like Facebook and LinkedIn across sites.

Firm implications

Openly accessible developer products such as APIs can be beneficial for companies but can expose a high degree of risk to the business if not managed and governed properly. A recent study by Benzell, LaGarda, and Van Alstyne (2016) shows that firms benefit from opening up APIs and building developer platforms through higher sales and higher market capitalization than their peers. The study additionally highlights that firms
are increasingly exposing data and services around their core through APIs. As APIs expose a firm’s services and, potentially, data, firms introduce terms of service to limit the possibility that external developers will use the openness of the developer platform to introduce a competing product or service.

Violations of such terms can lead to sanctions that include rate limiting or entirely discontinuing service to specific partners. For example, Uber shut down the price-comparison app Urbanhail, which compared Uber’s prices to competing ride-sharing services, as price aggregation and comparison are not permitted with the use of Uber’s API. Similarly, Twitter excluded video-livestreaming service Meerkat from using Twitter’s API broadly as their product gained major traction in the livestreaming space in 2015. At the same time, Twitter finalized an acquisition of Meerkat-competitor Periscope, which it later used to build out its own livestreaming capability. In both cases, firms are acting in order to protect their core business from potentially negative side effects resulting from their openness on the developer platform side (Darrow, 2016; Lomas, 2015).

In the case of Twitter, the company’s partially nontransparent approach to rate limiting popular applications that were close to Twitter’s core user experience and had gained traction among Twitter’s core users has negatively impacted its relation to third-party developers for years. At Twitter’s 2015 Developer conference, the company’s CEO Jack Dorsey openly apologized to developers: “Our relationship with developers got confusing, unpredictable. We want to come to you today and apologize for the confusion” (Koh, 2015).
**Governance need + Research question**

Given the multidimensionality and multi-tiered system effects on developer platforms, governance mechanisms and management of those platforms become key to successful deployment. This study examines the essential management decisions and governance mechanisms related to external developer platforms. This topic is studied in relation to the respective architecture and product development organization around developer platforms as well as the approach firms take to prioritize the respective level of openness.

**Methodology**

To synthesize product development and governance insights, the literature review in Chapter 2 introduces key concepts around platforms and governance structures on the firm level after defining the relevant technologies that make up developer platforms. Chapter 3 introduces the approach to the two case studies of this research. Subsequently, Chapter 4 presents the Akamai and Uber case studies in detail and discusses the findings. Chapter 5 then concludes those findings.

**Chapter 2: Literature Review**

**Underlying Technologies of Developer Platforms**

An API, or application programming interface, is a link to data or a service through a set of routines, protocols and tools that standardize transactions between multiple parties by giving defined access to an associated database or program. APIs are code and
serve as contracts between different parties (Woods, Jacobson, & Brail, 2011). While APIs themselves do not represent legal contracts, developers are oftentimes required to accept legal agreements such as terms of use, end-user license agreements, developer agreements, or service legal agreements to access an API service. These legal agreements regularly include API usage limitations and requirements, service-level expectations, and liability terms. The API provider is responsible for providing and enforcing the legal agreement (compare Legal Agreements of Developer Platforms, e.g. (Akana, 2017).

As such, APIs entail the format and type of communications between an application and the associated database or program. This method of communication does not require any knowledge about the internal workings of the associated database or program, which makes it agnostic to the source of the call (Benzell et al., 2016).

APIs serve an important architecture and governance role. Access to a defined set of resources can be precisely metered, and access permissions can be granted to a specific set of users. Thereby, adherence to firm-specific terms and conditions can be implemented to define economic and technical terms of use. These are foundational characteristics for establishing platforms (G. G. Parker, van Alstyne, & Choudary, 2016).

Open APIs are a class of APIs that allow third parties access to internal resources. They serve as the foundation for developer platforms, as firms can make data or services available to the general public or to a select set of partners to build additional services on top. Prior to APIs, firms leveraged electronic data interchanges (EDIs) to exchange data with other firms in one-to-one interactions from the 1980s onwards. APIs provide a
more cost-effective manner than EDIs do to communicate between parties (Benzell et al., 2016).

While APIs make up the foundation of developer platforms, a set of other products are often combined with them to provide a more compelling offering for developers. Additional products include software development kits (SDKs), integrated development environments (IDEs), libraries, deep links, buttons, and widgets.

According to Stackoverflow.com (Williams, 2014), an SDK is a client library with data files and sample code that supports developers in writing code to access a specific API or set of APIs. Client libraries are code samples which help developers perform a certain action without fully writing the functions themselves. In the case of Uber’s developer platform, SDKs can be a small button that gets integrated into the app, a widget that mirrors the entire booking flow and includes visual assets, or a raw client library that only includes the transactional components of the user flow. An IDE is a text editor with embedded functionality to compile, develop, and debug applications. Examples are Eclipse and Visual Studio. In addition, numerous developer platforms also include deep links as a product. Deep links do not include visual assets and link into the process flow of a service, for example.

Platform Ecosystems
Developer platforms go beyond API portals which represent one-directional interaction locations where developers access a firm’s resources. Developer platforms expand this offering to a multi-directional interaction pattern. Noori and Weiss (2013) describe platforms as a technology, product, or service that serves as the foundation for third
parties to develop new products that are complementary to the initial offering. Such platforms can be owned by a single actor (e.g., Apple’s application development ecosystems commonly known as the App Store), or ownership can be shared among a group of users (e.g., the Eclipse software development platform). For a thorough discussion of platform ecosystems, see Eisenmann, Parker, and Van Alstyne (2008), Tiwana (2014), and Parker et al. (2016).

Parker, Van Alstyne, and Jiang (2016) outline the value of open developer platforms. With broader access to the APIs or related products for developers, the open platforms can be leveraged as a resource for novel projects. Opening developer platforms shifts value generating activity from inside the firm to third parties outside the firm. This transformation results in a positive profitability impact in the longer term for the API providers. This effect is shown by Benzell et al. (2016), as data from the API Management solution provider Apigee demonstrates that developer platforms have a positive impact on operating revenue and market capitalization growth. The authors explain this phenomenon as resulting from the ability of developers to create a range of third-party applications that are complementary to the API provider’s core product, thus extending the reach. By moving from a developer portal to a developer platform, the API provider achieves usage of its developer products among an intensely used ecosystem of applications. This can further be facilitated by the API provider through increasing awareness for those third-party services. As a benefit to the provider, usage of its developer products by a community of developers fuels engagement with the firm’s services and potentially drives revenue.
Despite the generally positive impacts of openness, developer platforms face a general tension between openness and exclusivity. An open platform offers self-service functionalities, a low barrier of entry, and a leveled playing field for actors of all sizes. The objective is to facilitate an increase of innovation that allows the company to identify rising players or trends in the industry early before the increase to a significant size. By identifying uprising startups, a company can, for instance, identify promising acquisition targets at an early stage to broaden their offering.

Risks around openness include a loss of direct user interaction. By enabling third-party applications to embed core services like -- in Uber’s case -- requesting a ride into their application, it is up to the third-party service, such as the hotel building this integration, to design the core user experience and maintain the direct customer relationship with travelers using transportation services during their stay. Companies that open access to core product components beyond a purely informational character lose a level of control to directly steer the user experience. As companies generally aim for full control of their user experience, they can accept this tradeoff if it allows them to gain broader reach and scale with their product (Boudreau & Lakhani, 2009; Tiwana, 2014).

**Product Development Process**

The characteristics of platforms and related governance requirements also surface in the product development process. The approach that a firm takes for evaluating, building, and operating new product features directly relates to its governance and, therefore, its approach to managing risk on the platform. The API Management Framework (XapiX, 2017) lays out various life-cycle components of API management. During the exploration phase, the framework outlines the importance of a firm’s strategic
decision around openness and exclusivity. The strategic direction, alongside the competitive and legal risk assessment as well as the identification of data ownership structures within a firm, lays the groundwork for the governance framework. Figure 1 below illustrates the multiple stages of the API management framework on their links.

[Lean API Programs]

Start → 1 Exploration

5 Insights → 2 Development

4 Operations ← 3 Launch

Figure 1: XapiX (2017), Lean API Program, retrieved from www.xapix.io. Used with permission.

Architectural decisions during the development phase, such as the establishment of check mechanisms and governance boards, add further building blocks to governance around building external developer products. During the launch, specific requirements from external partners can influence the setup of specific security and process requirements. This can introduce new requirements to the internal development team at this stage.
Gartner (2016) identifies a strong correlation between governance and the product requirements process, which is an essential component of the product development process (PDP).

With regards to PDPs, Unger and Eppinger (2011) provide a comprehensive overview of various approaches. The risks during the development process that governance aims to constrain include technical (i.e., specifications for form or function of the product), product-market fit (i.e., dependence on customer demand), budgeting, and schedule risks (Keizer & Halman, 2009). Unger and Eppinger (2009) further outline that PDPs manage risk through reviews that are ideally defined as part of a firm’s governance. Those represent development gate checks between various stages with the goal of increasing the adequacy of the overall process.

Staged and spiral PDPs are two core approaches that can be identified:

Spiral PDPs repeat regular iterations regarding idea and concept development, detailed and system level design, as well as testing and integration. Overall, the spiral model follows the design pattern of agile development. Through the process, associated risks and governance requirements can be evaluated at an early stage of the development process, which proves to be cost effective (Boehm, 1988; Reich & Paz, 2008). In contrast, the spiral process comes at the disadvantage of increased requirements for management attention and potential delays for highly complex system integrations (Unger & Eppinger, 2011).

Staged PDPs are, with the exception of the initial design stage, sequential, separate, and one-way in nature. This model is often depicted as a ‘waterfall’ process (cp.
Erdogmus & Williams, 2003). In this model, iterations after individual stages have been completed prove to be costly and difficult, as they resemble a backward step in the development process. In addition, the staged processes have longer lead times and limitations regarding their rigidity (Anand & Kodali, 2008; Biazzo, 2009).

Two additional organizational limitations in product development highlight the importance of governance. First, the existence of organizational boundaries between teams requires process around involving relevant stakeholders (Ulrich, 2004, p. 24). Second, specialized teams that are involved in the development process have strong communication barriers across the team boundaries. Governance processes can help overcome structural barriers to ensure alignment (Tushman & Katz, 1980).

**Governance**

Governance is a key component for firms to align their goals around platforms, their architectural choice of technologies, and their approach to developing products with risk-limiting measures. Tiwana (2014) emphasizes the importance of governance for the vitality of an ecosystem around a platform. Misalignment of governance with key platform decisions around architecture and business model can prevent an ecosystem from growing or even cause its failure.

Governance is described by Miriam-Webster (2016) as the process “to control, direct, or strongly influence the actions and conduct” of one or multiple people. Calder (2005) similarly highlights the support of a firm’s strategies and objectives as the primary role of governance within organization. The tools of effective governance comprise “framework for the leadership, organizational structures and business processes, standards and compliance to these standards.”
In the context of information systems, Gartner (2016) outlines four primary functions of governance within a firm: First, the definition and allocation of decisions and accountability in accordance with the company’s business objectives; second, the allocation of funds in support and alignment of those objectives; third, the establishment of accountability measures for compliance with decisions and policies; and lastly, risk management within the firm in order to ensure that processes and actions account for consequences. The Organization for Economic Cooperation and Development (OECD, 2014) emphasizes the importance of identifying the role of a firm’s governing body and to align with that role as an organization.

In support of these core functions of governance, Weill and Ross (2005) identify three structural components of governance mechanisms: first, decision-making structures within the organization such as established roles and organizational committees; second, the existence of alignment processes to ensure widespread involvement in decision-making; and lastly, formal communications that refer to the process of informing stakeholders within the firm about changes. These three elements are core to internal governance structures.

Calder (2009) outlines external drivers for IT governance to emerge. His list comprises organizational diversification designed to maintain or achieve a competitive advantage in dynamic environments, emerging general IT governance requirements from regulators in OECD countries, specifically with regards to information and privacy, the increasing number of attacks and threats to information and intellectual assets that drive information security requirements, as well as the project governance needed to align organizational business goals and objectives with technology projects.
Governance structures within a firm can further differ by type. In their analysis of IT governance structures of 256 companies, Weill and Ross (2005) differentiate centralized, blended, and decentralized models of governance. This distinction is in line with the governance definitions of other authors, e.g. Noori & Weiss (2013). In their study, Weill and Ross (2004) relate the focus of firm performance to the three governance models and related IT governance mechanisms. As depicted in Figure 2, centralized governance structures promote profit maximization, while a growth focus is linked to decentralized structures.

![Figure 2: Weill, P. and Ross, J. (2005), IT Governance: How top Performers Manage IT Decision Rights for Superior Results, Harvard Business School Press, 2004. Used with permission.](image-url)

In relation to open platforms and collaborative communities, Boudreau and Lakhani (2009) note that governance is oriented toward norm-based interactions that follow
traditional social interaction patterns and is, therefore, generally informal. Noori and Weiss (2013) add the importance of regulatory elements as part of the governance toolkit. Their research finds that such regulatory elements further support the growth of the platform ecosystem by promoting the development of third-party applications and additionally controlling the quality of those integrations. One dimension of such regulatory elements is pricing models that can create a barrier to entry to some members of the developer community and thereby attract an, on average, more invested community base. Another dimension includes a range of developer products, or toolkits, that combine development frameworks with specific software infrastructure and lower the required effort of developers to integrate specific solutions. Finally, the authors mention sandboxes as a regulatory tool for third-party users to efficiently test their integrations within an actual deployment environment. This approach can serve as a means of testing the integrations of third-party developers before moving them to general availability.

Tiwana (2014) highlights the importance of effective governance of ecosystem platforms. While command-and-control structures of established firms rely on hierarchical authority of managers over the firm’s employees, this level of direct authority is not existent in a platform ecosystem. Developers on a developer platform act as free agents instead of employees, and they typically specialize in niche domains adjacent to the platform owner’s domain. Williamson and Meyer (2012) therefore argue that platform providers should rely on performance-based rewards to shape and influence the ecosystem rather than on command-and-control with punitive penalties.
In the context of ecosystems, governance provides the “context” in which innovation distributed among developers through novel applications founded on the platform can surface. Here, governance can be understood as the mechanism through which a platform owner guides and influences developers of a platform (Schilling, 2012, p. 159). It furthermore serves as a means to reduce behavioral complexity. This role is comparable to the role of platform architecture in reducing structural complexity around the platform.

Tiwana’s (2014) model identifies three dimensions of platform governance: decision rights partitioning, control portfolio, and pricing policies. First, decision rights partitioning refers to the allocation of decision rights, and, thereby, authority, between platform owners and users (i.e., developers). Second, control portfolio encompasses a collection of mechanisms through which the control of platform owners can be exerted over its users. This component includes the decision on which applications can be accepted or rejected. Evans and Schmalensee (2007) propose that these control mechanisms are a set of tools necessary to implement and enforce rules that support, reward, or punish specific usage patterns. Such tools include formal gatekeeping mechanisms, process controls, and metrics-specific reward or penalty structures (e.g., a minimum amount of revenue generated through the channel). Third, pricing policies, including the division of proceeds from the platform and financial incentives, are another important governance dimension. Incentives of platform owners and developers must be aligned to ensure that developers make the personal investment of working with the platform and building a business on top of it.
In aggregate, the governance dimensions rely on an “overarching collective goal for the platform ecosystem, [...] a sort of shared identity that defines the character of the platform ecosystem” (Tiwana, 2014, p. 125), which aligns with the goals and incentives of developers on the platform.

Overall, the field of research that applies the governance mechanisms to developer platforms is so far primarily untapped. Existing research in the field of governance specifically to APIs, a subset of developer platform products, focuses primarily on the technical layout of APIs in order to ensure functional, behavioral, and contextual alignment within firms (cp. Krintz et al., 2014).

Gartner (2016) further outlines that governance emerges and is established along the requirements process. This embeds the governance process into the broader product development framework.

Chapter 3: Case Study Methodology

This research is based on field-based case studies at two for-profit organizations in the technology space: Uber Technologies, Inc. (“Uber”) headquartered in San Francisco, CA and Akamai, Inc. (“Akamai”) headquartered in Cambridge, MA. For the theory building research, a knowledge-based perspective is applied, and the two developer platform approaches at the technology firms are compared. Based on both case studies, a set of propositions is defined. Both firms are comparable in their dependence on software to deliver their product and their organization size. Differentiators between the two include firm age, internal firm structure, and role of APIs. The qualitative and
deductive research process builds on Eisenhardt's (1989) process of theory building from case study research.

At Uber, field interviews were conducted as follows:

- In the summer of 2016, exploratory interviews were conducted with the leadership team of the developer platform team. The purpose of these interviews was discovery of the goals for the Platform Team within Uber and identification of general challenges relating to the governance and management of the platform. This phase was conducted on-site over the course of ten weeks.

- Between September and November 2016, three semi-structured interviews were conducted with the Product Manager of the Developer Platform team, the Engineering Manager of the Developer Platform team, and the Engineering Manager of the Marketplace Platform team.

- Public information about the Developer Platform team was collected, including a team profile on Uber's blog, presentations at API Days in Sydney and Paris, the team's Medium and Twitter sites, and articles in TechCrunch and San Francisco Chronicle.

To protect Uber's confidentiality, all non-public information has been removed from the thesis. The sections on the firm's internal product development process and team steering / KPIs have been removed prior to publication.

At Akamai, field interviews were conducted as follows:

- In September 2016, one exploratory interview was conducted with the Vice President of Developer Marketing at Akamai to discover the goals of Akamai's
developer platform teams and to identify initial differences between the two organizations.

- Between October and December 2016, three semi-structured interviews were conducted with the Vice President of Open Platform, the Vice President of Developer Marketing, and the Vice President of Information Technology/Chief Enterprise Architect, who has full responsibility over the internal API governance process.

- Public information from Akamai’s website, Reuters as well an interview with the Vice President Open Platform on A Total Disruption (Scobie, 2015) was reviewed.

- Internal documents describing the product management roles as well as process, functioning, and guidance for API development and governance were examined.

The semi-structured interviews followed a four-step process:

- Step 1: Identify developer platform set up, goals, and objectives. By interviewing product managers / Vice Presidents of the team who set the product agenda and who have a deep understanding of the architecture of the team and product, the overall mission and vision of the teams as well as the firm’s organizational structure were identified.

- Step 2: Identify and map out the product development process and internal governance including stage gates and checks as core elements of risk management measures. By interviewing product managers and engineering managers, both from inside the team as well as other internal stakeholders, who
are responsible for the entire product and feature lifecycle, an understanding of the internal processes and their implementation was developed.

- Step 3: Identify external governance processes. By interviewing product managers and engineering managers from the team, core processes put in place to ensure external governance and their approach to managing external partner relations were identified. In addition, publicly available terms of use were reviewed.

- Step 4: Identify developer platform team key performance indicators (KPIs) and the potential impact of governance structures on those. In all conducted interviews, we asked for KPIs for developer platform teams.

The four-step process was followed along the sections and individual interview questions laid out in the following. A detailed overview of leading and supporting interview questions is amended in Appendix 1: Interview Questions.

Company Introduction

This section introduces the companies and their business models and gives a brief overview of mission, vision, and corporate values as they relate to the management and governance decisions of the companies.
General Introduction

The missions and visions of the developer portal teams are introduced in this section, including their development within the organizations and their core business objectives. This section also refers to the general philosophy of the companies regarding openness.

Organizational Structure

The guiding question to gather information on the organizational structure is:

*What is your general setup/organizational structure?*

With this guiding question and a set of questions outlined in Appendix X, the overall strategic importance and potential impact of the team can be evaluated. This question also served as the basis for the organizational chart. The understanding of the organizational structure is important as it relates both to the structure for internal decision making and relevance of external partner relations and their respective importance.

Strategic Product Development Evaluation Process

The guiding question to gather information on the strategic product development process is:

*How do you evaluate the product roadmap internally?*

This section identifies the role of developer products, especially of APIs, for the company. This includes a brief introduction into the general architectural framework.
within the organizations that both internal and external governance emerge from.

Furthermore, key drivers for product roadmap and feature decisions are identified as they relate to the specific roles stakeholders in the following two sections.

**Internal Governance Process & Risk Management**

The guiding question to gather information on the internal governance and risk management is:

*How do you govern the platform internally?*

The internal governance process is evaluated by analyzing the decision-making structure and the internal alignment process. An internal process chart provides an overview of the decision flow and stakeholders involved in the process as well as related measures to ensure the accordance with the structure. In addition, the alignment process analysis identifies formal and informal governance mechanisms to ensure that all relevant stakeholders are involved in the process.

**External Governance & Partner Relations**

The guiding questions to gather information on external governance and partner relations are:

*How do you manage partner relations? And how open is the platform?*
The external governance process is evaluated regarding legal framework for usage of the developer products by partners. Transparency on the company’s roadmap, strict control of terms of services, and openness are evaluated as guiding indicators for the type of governance the company chooses. This evaluation is done through analysis of general terms of use for the developer products, individual agreements with partners, and the communication style and measures that the company implements.

Team Steering / KPIs

The guiding question to gather information on the metrics for team steering is:

*What are the key performance indicators for the developer platform team?*

As a final research component, the performance metrics for the developer platform teams are analyzed. The understanding of those measures provides an indication of how the teams are incentivized. Specifically, the section is targeted at analyzing whether, and to what extent, considerations around effective governance and management of the platform play into the team objectives.

**Chapter 4: Case Studies**

Chapter 4.1: Akamai, Inc.

Company Introduction

Akamai is a global Content Delivery Network (CDN) with the vision of “making the internet fast, reliable, and secure for its customers” (Akamai, 2017). The service offering
comprises solutions for web performance, mobile application performance, media delivery solutions, and cloud security.

The software company with more than 6,600 employees is headquartered in Cambridge, MA, USA with revenues of 2.2bn USD in 2015. The company's core product, a content delivery network, is among the largest distributed computing platforms. It served 15-30% of all web traffic in 2015 (Tharakan and Patnaik, 2015).

Core to its operation is a global network of servers for which it rents out capacity to its customers. Having servers in the vicinity of their users allows the customers of Akamai to make their websites function faster. Technically, a user of an Akamai customer gets redirected to one of the website's copies on Akamai's servers when a user navigates to a URL of an Akamai customer. The company uses a range of self-service tools and dashboards to deliver additional analytics and insights to its users.

Introduction of the Open Platform Team

The developer platform team was formed under the name *Open Platform Team* at Akamai in order to increase the functionality that the company offers to existing customers. Driven by the need of its customers to establish "deeper integration into existing processes and operating environments" (Scobie, 2015), Akamai started providing a set of modern REST-based APIs to its customers. This API allows their users a higher degree of control and customizability to retrieve insights around the usage of Akamai's services. Similarly, it allows Akamai to extend its product offering to a wider audience. Going forward, developers of companies that are much smaller than
the usual customer base of Akamai will soon be able to access part of the company’s products and services through free starter packages.

Overall, the company aims to thereby “create more opportunities for community collaboration and innovation, and provide deeper visibility into, and greater control of, how enterprises interact with Akamai technology” (Scobie, 2015) -- both for future and existing customers.

In addition to the APIs that are available on the Open Platform that allows users to configure meta-information and retrieve usage analytics, a second category of APIs is provided on a partner-specific level including customer, invoicing, and billing information.

Organizational Structure

Within Akamai, the Open Platform team is led by Corey Scobie, Vice President of Open Platform and Product Experience, and Nick Tran, Vice President of Developer Marketing. Both report to Rick McConnell, President of Akamai, who reports directly to Chief Executive Officer Tom Leighton. The team represents the external facing component of Akamai’s current transformational change to API-fy the organization.

The VP of Open Platform and VP of Developer Marketing jointly lead a team of 219, split between Product Management (5), Developer Marketing & Relations (4), User Experience Design (40), Engineering (160) and Program Management (10). In addition, the team around Developer Marketing & Relations is planned to grow from four current
roles in marketing, community, and support into a comprehensive team of 25 as part of an effort to further promote the Open Platform initiative.

The new team structure will include teams for developer evangelism, marketing, events, content, engineering for the website, a support team that works tactically with developers, and developer partnerships.

In addition to the open platform team, a second, internally facing team exists and primarily serves the internal community. This team is located under the CIO and, in contrast to the Open Platform team, is not part of the product organization. This internally facing team focuses on APIs around customer-specific information such as invoicing and billing, which are not provided through the open platform. This set of more restricted APIs serves large partners such as Microsoft Azure.

For the internal API development process, a multi-functional team is usually involved, covering roughly 35 people at the core across the roles of Product Management, Program Management, Services & Support, Product Marketing, Platform, Product Architect, IT Systems, and Product Engineering. In addition, the broader circle includes further engineering and business functions such as ordering or billing increasing the group of involved stakeholders to 60-70.

Strategic Product Development Evaluation Process

Akamai was previously -- and to a large extent still is -- reliant on a set of 150 apps without discrete API interfaces that were built without the aim of allowing Akamai customers to directly integrate with the underlying services. Those legacy applications
represent siloed, full-stack apps that were developed in parallel to APIs. As part of a large architectural shift, the company has created a new platform as a service application development architecture that relies on APIs. This focus requires developers to separate rendering layer and business logic, which overall creates a more flexible development environment for Akamai. The internal processes for API development reflect this by requiring specific stakeholder engagement as laid out in the following.

Akamai’s overall focus is on building user interfaces (e.g., dashboards) and APIs that go along with its entire product offering. Over a timeframe of 18 months, the company is on a mode to “api-fying everything” (Corey Scobie). The company is approaching this by promoting openness for the overall developer platform:

“We want it [access to the platform] to be as open as possible, we want to be inclusive, not exclusive.” (Nick Tran, 2016, Thesis Interview. Used with permission.)

In order to achieve a higher demand for the developer products and a higher degree of openness, Scobie sees the team going through three core phases:

“The API-fication at Akamai has three discrete stages: Currently we are in the first phase of our evolution with the goal of getting interfaces out. The second phase is about rationalizing the portfolio and creating new APIs where a business case exists. The third wave will be about canonicalization to simplify the interaction model between APIs.” (Corey Scobie, 2016, Thesis Interview. Used with permission.)
Supported by leadership of the Executive team, the company as a whole has established an architectural model to require developers to create their own APIs for new services. This structure has so far led to the creation of 150 discrete API endpoints in an effort to share API interfaces with partners. While there is no specific dependency diagram in place yet to map out the evolving internal API ecosystem in detail, the overall process is guided by Akamai’s annual interlock process. Specific APIs in that context are strategically prioritized, such as the property management API that exposes core functionalities and is among the more powerful APIs. Akamai generally tests specific APIs in partnerships first while they are in beta before making them available to the general audience. This process is openly communicated on their platform.

On the Open Platform, Akamai offers a set of developer products including APIs, SDKs, and libraries for several languages. General prioritization of these products is driven by the requests from business partners through the executive level or the business organization as well as the direct interaction of the Open Platform team with developers, (e.g., in the support forum or at conferences).

**Internal Governance Process & Risk Management**

Akamai internally applies a process referred to as PACE (Product Agility, Consistency and Excellence) to manage and govern the development of new products. Akamai employees can inform themselves on an intranet site about the PACE process and its goals.

Overall, the process comprises the following elements:
- A cross-functional setup with program core and extended teams with responsibility for program execution
- A structured development process framework that provides the complete set of cross-functional activities starting from concept stage and laying out the key steps towards launch
- An Executive review board (Product Review Board) that governs and guides product development internally
- Internal phase reviews as part of the decision-making process
- Phase Contracts between the core team and the Product Review Board that specify scope and boundaries of core team discretion

Overall, Akamai’s internal guideline on PACE specifically refers to three goals:

1. Enable Cross-functional alignment for successful product commercialization
2. Provide executive visibility into and a governance model for Product Development Efforts
3. Provide structure to product teams along a structured stage gate process outlining tasks, steps, and deliverables to help product development core teams execute efficiently on their planned efforts
The APIs are generally developed along Akamai’s core product development process with the stakeholders outlined above.

(1) Decision-Making Structure

The Open Platform team strategically aligns with the other core product lines, primarily the core functions of Akamai’s Web Division led by Rick McConnell. As part of achieving alignment, Akamai has established a governance group, referred to as Core Interlock Group. This group is the interacting platform between product, engineering, security, sales, and finance teams. To guide the overall product development, the group sets and signs off priorities on a regular basis. Depending on the required output and interaction level, this sign-off can be a quarterly, monthly, or project-based meeting.

The Open Platform team lays out a specific funnel process on how specific APIs are potentially improving client as well as developer relationships and shall therefore feed into the overall product.

On the technical implementation side, Akamai made use of short sprints for multifunctional teams to ensure full alignment on a work level. In a two-week phase, preparation needs and design requirements from partners and developers are gathered and aligned with the overall strategic direction and timeline with the involved business units. During the five-day sprint, the cross functional team scopes out 15 APIs and produces production code that is then handed over to the regular quality assurance process.
As a sub-team of the interlock governance process, a usage discussion team meets on a weekly basis to review requirements and audit usage. This team can manage access control and enforce rate limiting.

In addition to external APIs and developer products made available through the Open Platform, the governance process also captures internal APIs. Internally, Akamai leverages an API catalogue to support transparency over existing APIs. The APIs that are not yet provisioned through the Open Platform but are desired to be implemented there are prioritized and made available through the internal interlock governance mechanism. The Interlock Governance team sets priority on an annual basis and assigns human and monetary resources. Access to the interlock mechanism is on a need-to-know basis with Akamai’s business partners on an account level (i.e., contract level). For those closed arrangements, a one-to-one contractual relationship, including a specific set of Service Level Agreements and implementation requirements, governs the usage of Akamai’s business relationship.

The establishment of the interlock team with a strong focus on guiding resources for API development is the result of an accumulated number of requests driven by customer demand for new APIs, which led to the need for stronger prioritization.

(2) Alignment

To further ensure security and effectiveness of design, Akamai has created an internal “API Working Group” whose mission it is to “provide Akamai guidance on how to design world class” Web APIs and expose common design flaws. The working group is the guiding structure within the company to develop standards, conventions, and guidelines
in core areas of API development (e.g., API Design Best Practices and API Design Principles, API Consistency and Ease-of-Use, Hypermedia-Driven APIs, API Specification and Documentation, API Testing Patterns, API Design Review Approaches). The scope of the working group comprises RESTful APIs, internal APIs, and external facing as well as Open APIs. SOAP APIs and the use of specific programming languages, tools, or frameworks for API creation are out of scope.

With respect to Open APIs, the working group references internal security conventions and directs them to the Open Platform team.

In support of those working groups, the use of specific tools such as Apiary, a tool that allows modelling of APIs and receives outside validation, was targeted within the organization. The use of this tool, however, did not achieve the envisioned level of traction and, therefore, more guidance is given back to the individual working groups that govern the review process and include architectural reviews. Within Akamai, about 400 engineers are directly involved or impacted by the guidance of the working group.

To manage risks, best practice sharing is key when it comes to processing knowledge about API design throughout the organization. The working group at Akamai facilitates this process through an API Best Practices overview. An excerpt of this is depicted below.
IX. OPEN API Constraints

This section provides information on special constraints that must be considered for OPEN APIs (re: https://developer.akamai.com).

IX.A. Message Limitations

API designers who create "OPEN APIs" that use EdgeGrid...

- SHOULD be aware of the Conventions for Edgegrid Conformance. These conventions result from known limitations regarding the use of ESSL Ghost.
  - For example, as of August 2014, limits have been placed on the Client Authentication with respect to body hashes on the signature whenever PUT or PATCH are used
  - SHOULD consult with OPEN Platform engineering or product management if they have any questions.

In addition to the working groups, the previously-outlined PACE process further supports alignment among stakeholders for the development of specific product for the Open Platform team. Mainly driven by the respective product managers, all relating functions including IT Systems, Platform, and Product Marketing are included in the process to ensure alignment. A stage-gate process with dedicated Product Review Boards and Executive involvement help reach cross-functional alignment. The first review is implemented after the concept assessment phase. The second one is following the product definition and planning phase. The development and beta testing phase concludes with a review around launch commit.

External Governance & Partner Relations

Through business development, the needs of traditional partners are brought into the organization and evaluated against current service level agreements and offerings. In addition, the Open Platform identifies needs on a broader developer level, which becomes part of Akamai's evolving user base. In addition to business development, the
leadership at Akamai plays a vital role in gathering requirements from CEOs of customers.

There is generally a large push for openness on executive level, and afterwards, on the operational level, the internal interlock governance mechanism comes into play to define the details. To achieve the goal of creating a stronger level of openness, Akamai aims to extend its current offering to developers by adding free accounts with potentially instant sign-ons.

(1) Terms of Use

Akamai's customers agree to individualized terms of service as part of the establishment of business between the two parties. These are specific to their core Akamai account in the LUNA platform. To date, access to developer products requires users to be existing customers of Akamai's services. In addition, users accept terms of use that are specific to the use of APIs. The API-specific terms of use include a reference to the LUNA account. In addition, the terms limit the scope of information usage. In contrast to Uber, Akamai does not grant usage of the API data in any third-party products. This constraint highlights the informational component of the API. Users are prevented from building third-party products that enrich Akamai's offering.

(2) Individual Agreements

Akamai's current offerings have a limited number of APIs available to the public. As of December 2016, 38 APIs are listed on the Open Platform (for a complete list see
Appendix 4: Akamai API Overview) out of the roughly 150 APIs that are internally developed. For the usage of the APIs, an existing business relationship is currently required. The individual contracts between Akamai and its customers lay out specific terms of use and service level agreements; they are not generally applicable to partners. The API services are currently not offered as a standalone product and provisioned as part of an ongoing business relationship where it is bound to a customer identity.

(3) Communications

Customer communication around the Open Platform is facilitated primarily on two levels. On the one hand, direct customer communication is driven by business development, which, as targeted at individual partners, is the primary, established channel at Akamai. In addition, it is facilitated by Akamai's outward-acing community platform, which is part of the Open Platform targets a broader audience. The company provides generic guidance on its community forum as well as in its Learn Portal (accessible through https://developer.akamai.com/learn/) and on Github (accessible through https://github.com/akamai-open/).

Team Steering / KPIs

To ensure effective steering of the initiatives around the developer platform, KPIs are tied to the maturity stage of the API maturity lifecycle. In the initial launch phase, adoption is a key measure for the team at Akamai, measuring how many people are
aware of the API and how many customers start using it. At a later stage, value and usability become more important than adoption. This metric can be measured by the material value the API generates for a business, time to implementation and a customer effort score that captures the complexity of tasks and the quality of the user experience.

On the developer marketing side, top level metrics include active developers, number of developers in the program, and net new signups. The outreach activities can be captured by measuring the number of articles published, number of developers reached, and the number of developer support cases closed.

A set of metrics that is tightly coupled with the external communications component of partner relations include KPIs set forth in the service level agreement. These KPIs include frequency of chance, uptime, and further availability metrics.

Chapter 4.2: Uber Technologies, Inc.

Company Introduction

Uber, a technology company founded in San Francisco, CA, USA in 2009, is an on-demand economy company that focuses on providing transportation solutions. According to Harris and Krueger (2015), the hyper-growth startup is responsible for two-thirds of all activity in the platform-based labor market. The company’s main product, a smartphone application, matches consumers seeking rides or deliveries (“riders”) and those providing these services (“driver partners”) and handles the payment transactions between those parties. Uber’s service has grown from an offering in San Francisco in
2009 to a coverage of 550 cities in more than 70 countries by the end of 2016 (Uber, 2017a).

To use Uber’s services, users download the app and create an account for free. When seeking a ride or delivery, the user opens the app and sees which transportation products are available at the location. Upon entry of the route, an estimate of time to arrival of the driver-partner and, for some products, also an estimate of price and overall travel time is provided. Available driver-partners are sequentially presented with the opportunity to accept the order until the request is served. The driver-partner who accepts the request picks up the rider or item to be transported and drives to the desired location (Cohen, Hahn, Hall, Levitt, & Metcalfe, 2016).

Introduction of Developer Platform Team

The mission of the Developer Platform team is to empower developers to build moving experiences. Through its open, public product suite, it aims to integrate and support apps with its transactional transportation services.

The developer platform team at Uber offers products comprising APIs, SDKs, and other integration elements such as buttons, widgets, and deep links on developer.uber.com. Launched in August 2014, the developer platform has since supported multiple new product partnerships ranging from Facebook Messenger to Amazon’s Echo.

Users of the platform include large industry players such as American Airlines and Hilton, as well as a long tail of developers who are building applications for niche markets or side projects. Overall, the Uber developer platform has a number of high profile partners for third-party integrations such as Google Maps (Mapping), Facebook
(Social Media), Hyatt (Hotel), United (Airline), Foursquare (Mapping), Zomato (Restaurant), and Transit App (Transportation), in addition to tens of thousands of users (Noonan, 2016; Uber, 2017b). Business managers use the platform to inform themselves about Uber’s offering as it relates to data integrations, and software engineers leverage it for gaining access to Uber’s data and maintaining those integrations. The core value that the products aim to support are an improved experience for users of third-party applications (e.g., by giving airline customers a direct and integrated form of ground transport when they are using the airline’s services) (Uber, 2016).

For Uber, the platform -- among others -- is a means of increasing its reach; driving up ridership, drivership, and deliveries; and positioning itself as a stronger technology brand among talent.

The Washington Post (Levy, 2016) has launched a special offering around the Trip Experiences API, an offering that lets third-party applications receive trip-specific information about an Uber user if they decide to authorize the third-party app. The Washington Post describes the partnership with Uber’s developer platform team as follows:

“Services like Uber are showing the way with their suite of APIs. It’s a meaningful opportunity to integrate them in a way that makes sense for our users. We know from user research that our readers tend to switch back and forth between The Washington Post app and the Uber app when they’re on an Uber ride. They like keeping an eye on their trip status and estimated time of arrival. We think it’s our responsibility
to provide that added context to make our app a more seamless news consumption experience. [...] To introduce and test this feature, in the coming months, we’ll drop the paywall for users on Uber trips during our initial rollout period. This means new and casual readers can read articles during their commute or on their way out for the night for free. This could be one of the best ways to truly gauge how a feature like this can help our readers.” (Levy, 2016)

While Uber has internally leveraged APIs for its mobile applications prior to the establishment of the Developer Platform team, the APIs were not designed for public consumption. Therefore, a dedicated team was built from the ground up (Noonan, 2016).

Organizational Structure

A team of more than two dozen people suggests a hybrid model at the intersection of multiple engineering teams where both internal teams as well as outside developers are seen as direct customers of the developer products. Key functions within the teams that the developer platform team regularly interacts with include business development, international growth, product development, and marketing. The team is headed by a product and an engineering lead (Uber, 2016).

The Developer Platform team consists of a Developer Relations team, API platform team, API features team, and partner engineers among others. Developer advocates from the team serve external customers in order to increase adoption of developer products externally but also to build a better understanding around the supporting role
of the team internally. Furthermore, partner engineers support external platform partners to support and drive adoption of the Developer Platform products among external partners. They are closely involved in the definition of the product roadmap with their close interactions to external customers and other groups within the Developer Platform team, such as API Feature engineers (Uber, 2016). The API feature engineers focus on building new product components for Developer platform. In addition to the feature engineers, the platform engineering team scales and supports the features that are being provisioned by the API features team and ensure uptime and performance of the developer products (Uber, 2016).

**Strategic Product Development Evaluation Process**

Decisions around new product features are largely driven in smaller teams. One of Uber’s values is *Let builders build*, which the Developer Platform team leverages internally and externally with their products. In result, the API features engineers on the developer platform team, for instance, own the entire cycle for new features, from the initial design phase to requesting feedback for technical proposals and the implementation and launch phase. The iteration cycles are fast, and the team ships code every day, which is made possible by small and fast production cycles. Results from the features team are directly tested with engineers from the platform team to ensure quality. For launch preparation around specific features, the engineers consult with the developer relations team and partner engineers who have the relation to the partner (Uber, 2016).
A key area of responsibility for teams within Uber is to ensure an adequate use of data regarding user activity, privacy, and competitive landscape. Uber follows the approach of giving their users full control over the use of their data in third-party applications. According to developer.uber.com, this control is enabled through a standard authentication process (OAuth 2.0) that gives users the control over their data flow into other applications to which they link their account.

At Uber, it's important that people are empowered to build. This helps define the necessary philosophy which is primarily driven by opportunity and abundance, not control and scarcity.

Internally, the Developer Platform team collaborates with other teams such as UberRush to leverage the technology to build user-facing functionality and scale partnerships. In its hybrid character, the team both directly serves developers and also relies on business development teams to steer those interactions with other partners (e.g., Lam, 2016).

The prioritization of features for the technology products is similarly based on internal and external signals. Externally, developer requests on Forums (e.g., Stackoverflow), events and hackathons (e.g., https://uberhackathon.devpost.com/) as well as the one-to-one interactions of partner engineers with developers serve as a source of inspiration (Noonan, 2016).

External Governance & Partner Relations

The role of the Developer Platform team at Uber is to "innovate beyond just the utility of connecting and extending functions of your product." By focusing on three core
principles -- utility, revenue, and distribution -- the team aims to spur innovation beyond the company borders. Developers create “not just an extension of our own product, we think of this as a developer platform that real business will be built on top of” (Noonan, 2016).

The first principle, utility, refers to access to the core function of Uber. Here, it is making transactional services available for others to implement in their apps and extend the product. A restaurant app can include a pick-up and drop-off service for its customers, for instance. A set of standards and terms ensure that the integration meets the minimum standards that Uber customers can expect, whether they are ordering from a third-party service or Uber directly.

The second principle, revenue, refers to the ability for developers to earn money with transactions they facilitate through the API. “If you’re asking developers to write apps using your platform, they have to be able to pay the bills.” (Wintrob, 2016). Uber’s affiliate program allows developers to earn 5 USD for every new US rider taking a paid trip. Additionally, the team is evaluating other options for payments to developers to further align incentives.

The third principle, distribution, refers to the ability of Uber, through its own brand and reach, to support third-party applications built on top of the platform. This principle can be seen in the showcase section of the developer platform site (i.e., developer.uber.com/showcase) as well as in joint publications such as on Medium (i.e., https://medium.com/uber-developers/tagged/partner-spotlight). The addition of trip experience APIs to Uber’s offering that allow developers to interact more extensively
with Uber riders during their trip and serve custom offerings is a further example of tools in this category (Wintrob, 2016).

The activities of Uber’s fraud team extend to the API integrations. This team monitors the activities around the Developer Platform products such as account creation. In terms of limiting partners, Uber does not shut down applications that have similar ideas as long as they comply with the terms and conditions, some of which are outlined below. “It’s not our job to pick winners and losers,” (Said, 2016) states the team’s product manager, though the team makes sure to have a “kill switch” (e.g., rate limiting or revoking access for third party apps) to enforce its terms of use (Noonan, 2016).

(1) Terms of Use

In order to avoid negative consequences, the aim for openness is secured through a set of terms and conditions that allow Uber to guide, and potentially shut off, certain integrations. Uber’s approach is to support a general push for openness with core features of exclusivity. Sales and business development teams have an advantage offering specific, novel integration to set up and support partnerships with large brands. This arrangement limits risks around brand dilution, reduces the likelihood of unwanted consequences, and increases the exclusive bargaining power. If possible solutions around the available developer products are sensitive, such as granting access to core data streams and services around Uber’s core products, the company makes integrations more exclusive, focused on few partners with which Uber has direct contact and potentially influence/leverage over regarding the individual integration.
A key measure of the external governance with partners of all sizes is Uber’s terms of use. Terms and conditions around the affiliate service program amend these core terms. They outline core terms around but not limited to general usage of the Developer Products, revoking access, data collection, use restrictions, user data, independence, and granting license.

To ensure alignment with its general strategic direction, Uber includes core restrictions on user data, competitive positioning, and monetization on the use of products built upon the API in the terms of use. While the company generally promotes commercial use of the API in third-party products, limitations on type of application and competitive comparison exist, among others.

Key areas of controlling the competitive framework and user experience are laid out in the area of restrictions to the users of the API and act as a key governance tool. An excerpt of those sections is paraphrased below (for a complete overview of the Terms of Use see Appendix 3: Uber Developer Platform Terms of Use):

- **Competition**: No inclusion or endorsement of third party services that compete with Uber’s products and services
- **User experience**: Need to clearly disclose upfront fare to users prior to initiating a ride, if applicable; no allowance to start or end a trip without consent of the end user; no allowance to modify the user’s pick-up or drop-off location
- **Data sharing**: Easy access for users to their data; no permission to share user data
• Data storage: No collection of data

• Communications: No allowance of communication that can be perceived as harmful to Uber’s brand

• Security: No allowance to use services to the extent that the use could negatively impact Uber’s servers or networks; all data must be encrypted and transmitted over a secure, encrypted channel such as HTTPS

• Business model: No allowance to charge users for the services that are made accessible through the API

• Design: Need to comply with Uber’s design guidelines

Uber has enforced its terms of use with several applications, including a price-comparison application UrbanHail that directly compared prices of Uber and its US-competitor Lyft for users. The company was warned about the violations by the Uber developer platform team at the day of launch and was asked to remove features that violate the terms of use. As UrbanHail did not react to Uber’s demands for changes, the deactivation of their user account was announced three weeks later and executed after another goodwill period to adapt the service (Benedelman, 2016).

(2) Individual Agreements

The nature of general self-service terms and conditions, which developers typically accept when they retrieve the allowance to use Uber’s Developer products, regarding integration, call volume, and reach as well as partner requirements, drives the need for individualized partner agreements at a certain scale (Noonan, 2016).
Part of the developer products have limited access. Developers can inform themselves about the offering on developers.uber.com where most endpoints are documented. Applications for further access or the addition into beta access programs is facilitated through a form on the site. Uber cites the necessity to learn “how these endpoints might be used” as a reason for granting access to a limited set of developers, such as in the case of the Driver API (Saad, 2016).

(3) Communications

The primary objective of the team beyond the core operational agreements is to ensure alignment between the directions of partners with Uber’s own development agenda. Communication plays a key role in avoiding breaking changes, deprecations, or partner-specific rate limiting, which are triggered by competitive or legal considerations around the use of Uber’s developer products. Such changes create existential fears among partners and can be avoided by communicating “early, often, and with high empathy around a timeline for deprecation [or rate limiting]” (Chris Saad, 2016, Thesis Interview. Used with permission.)

To ensure stronger alignment with Uber’s interest, the team communicates actively through its Twitter channel (i.e., https://twitter.com/UberDevelopers) and Medium publications (i.e., https://medium.com/uber-developers). The latter includes product announcements and provides a channel for partners to contribute stories about joint integrations on the Uber publication, or feature such third-party applications as in the case of the Trip Experiences API (Uber Developers, 2016). In addition, part of Uber’s technical communication is furthermore facilitated through Github (e.g.
https://github.com/uber/rides-android-sdk) and Stackoverflow, where the team actively addresses user questions.

Chapter 4.3: Discussion

Akamai and Uber vary in their life-cycle stages as companies and also in the way they approach the management and governance of their developer platforms. Table 1 depicts the differences along core dimensions.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Uber</th>
<th>Akamai</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>Founder-run</td>
<td>Founder-run</td>
</tr>
<tr>
<td>Maturity Stage</td>
<td>Hypergrowth</td>
<td>Steady growth</td>
</tr>
<tr>
<td>Primary Product Role</td>
<td>Distribute Product</td>
<td>Nurture Product</td>
</tr>
<tr>
<td>Philosophy</td>
<td>Mostly open</td>
<td>Mostly exclusive</td>
</tr>
<tr>
<td>Organizational Setup</td>
<td>Hybrid</td>
<td>Product</td>
</tr>
<tr>
<td>External Governance</td>
<td>Developer T&amp;Cs</td>
<td>Existing contracts from other products</td>
</tr>
</tbody>
</table>

Table 1: Company Comparison for Selected Dimensions of Developer Platform Governance

As the more established firm, Akamai has developed more rigorous governance of its platform. Planning cycles are longer, the overall product development process is
staged, and the current implementation is geared towards exclusivity and alignment with customer products that are tightly coupled to the API. The business needs and product development are steered through the interactions of leadership or sales with Akamai’s core customer base. From there, demands are passed onto the platform organization, which addresses them in a well-defined form. Standards, working groups, and review committees across the organization support the development and align with the overall guidance. As such, the developer products aim to extend the offering to current users. Currently, the company is going through a transition to introduce a new developer product that allows users who are currently not Akamai customers to open a new account and use Akamai’s services through the API in a limited form. This change represents a shift towards a new growth model and is supported by the top of the company by recognizing the need for further developer involvement.

“The future of the Internet lies in the hands of developers and architects.” (Tom Leighton, CEO of Akamai) - retrieved through the company’s blog Akamai, 2013

Akamai’s shift towards a stronger growth model with a new customer base precipitated organizational growth to build up the Open Platform team, especially through Nick Tran’s (VP Developer Marketing) organization, which is expected to grow from 4 FTE to more than 25 FTE.

Uber’s model shows that despite being deeply ingrained in the company’s values, the team is positioned in a hybrid model to both serve more internal (potentially exclusivity geared) directions such as business development and similarly focus on external users as the primary customer of the platform. The opinion on the team roles varies across
the level of leadership. Uber accepts this non-conformity as a virtue and puts the team in a position to prove its value, both in terms of staffing as well as its dual position to drive business serving other internal teams and external teams on their own. This is well aligned with the company’s value of *Let builders build* (Noonan, 2016).

Akamai is preparing to shift its closed, exclusivity-driven approach to a more open approach, which is addressed through additions on the Vice President-level and team growth in various developer relations roles.

I was asked “do what is right for developers” (Nick Tran, VP Developer Marketing)

It is unclear to what extent planning and product development cycles will be adapted given the stronger focus on smaller customers that potentially require faster feedback cycles. The company has laid a strong foundation through its working groups to ensure consistency in its offering, but current processes require the capturing of user needs from executives as well as the sales organization.

The focus on current customers is aligned with Weill & Ross’ (2005) findings for companies with a focus on profits. In its current setup, Akamai focuses on core competencies. Among others, key IT governance mechanisms include a clearly defined process architectural and capital approval process. Business processes are standardized, and the overall governance is more centralized. According to the study, this approach aligned well with the reduction of business process costs and a return on investment focus. With Akamai’s adjusted goals for reaching a broader range of developers and therefore growth focus, the governance model is likely shifting towards a more decentralized model. Given its process-driven nature, the company is not expected to rely heavily on the internal interconnectedness but rather will embed new or
further leverage existing review boards to allow faster feedback cycles from the “new” customer base.

In the performance model, Uber falls into the growth model that underlies the company’s position as a hyper-growth company, reaching a valuation of $18.2 million within three years and $68 billion within seven years since its founding (Philbeck and Työppönen, 2016; Sorkin, 2016). Local innovation by the team within the organization is highly encouraged, where it facilitates decentralized innovation among the broader transport ecosystem. The system overall depends largely on the overall optimism within the company, driven from the top leadership, which encourages decentral leadership and risk-taking (Noonan, 2016).

Both companies use terms of use to govern the use of the Developer products by external parties. In the case of Uber, the terms of use that are specific to the API are more extensive given the self-service nature of the platform. Akamai manages to keep its terms more concise, as they serve as an addendum to existing partner contracts. Overall, the terms and conditions for API and developer product usage are an important means for mitigating the use of the product beyond the boundaries that a company accepts. While Uber encourages innovation through the platform and wants other businesses built on top of the developer products to flourish, it manages to control the use of its APIs. This control highlights the terms and conditions as an important means for ensuring a consistent user experience across platforms while inhibiting competitive business practices and structures beyond an acceptable range for companies.
Chapter 5: Conclusion

Summary & Key Takeaways

Engaging external developers with the digital products and services of the firm is a goal for more and more enterprises, Akamai and Uber are just two examples. The two case studies show how differently companies approach this goal: different underlying goals, different internal product development processes, and different mechanisms to manage risks around the platform.

Firms must balance the benefits of openness with the risks of exposing data that can weaken their own competitive position. Identifying internal goals and guiding principles is necessary to give the developer platform team clear direction for their activities. To realize these goals, a set of tools, rules, and behavioral rewards can be implemented.

The overarching product development process that a firm applies shall further be aligned to those goals and mechanisms. Due to the agile character of interactions with developers (e.g. through emails, forums, and events), the spiral development process tends to support platforms with a higher degree of openness.

Once the primary goals and processes are defined, this research identifies several key success factors for platform deployment:

First, working groups and internal standards for API development, including best practices, serve as how-to guides for internal teams. They serve as an important internal governance tool to ensure alignment across teams as companies become more decentralized in their shift to turn their API portal into a more growth-spurring developer
platform, moving from a strong utility focus to a broader enabler role for other businesses. Furthermore, interconnected networks within the companies along with *small world ecosystems* (Iyer, Lee, & Venkatraman, 2006) support internal alignment.

Second, external contracts play an important role to ensure effective governance for partner relations. Terms and conditions represent a legal agreement between the developer and the API provider and additionally lay out the specific terms of use of the developer products. Specific requirements ensure minimum standards around user experience, brand and product design, as well as the third party’s freedom with regards to competitive scope and business model. Trust plays an important role in the API-powered partnerships, as external integrations are not fully controlled by the API-providing party, and, vice versa, the external party relies on the quality and integrity of a provider’s service as it invests in the integration of that service. Coherent, collectively exhaustive terms of use help outline the partnership both on a self-service basis for a large number of developers as well as individual contracts. Research shows that clarity and aim for completeness around terms of use serves as an important means in managing risks around external integrations. The strongest forms of external governance contracts are individual agreements with partners. These agreements give API providers the highest level of flexibility in adapting the terms to the partner-specific needs and also include the highest complexity in managing these contracts as they are not standardized to a level that general terms of use are.

Third, access limitations serve as a means to steer new application development among partners. Both Uber and Akamai limit access to specific services to a defined set of users and partners. This approach proves to be an effective tool to better understand
the system implications around new services that are being built and ensures tighter control of these partners. Such beta programs support the internal understanding of potential weak spots in the external governance, which must be addressed upon granting access for a larger community to the Developer products and tools.

Finally, open communication to developers further supports partner relations. By communicating a company’s own strategic roadmap along with suggested areas of innovation as well as potential no-gos, the API provider can help guide the resources and energy that partners are investing into their integrations. Companies with developer platforms should be aware of the dependence of their external partners on internal decisions that may alter the strategic path that a firm takes with its platform. Both companies presented in the case studies are well aware of the Twitter-effect, in which Twitter outraged part of its developer community by shutting down core services after the company had struggled to protect key value-adding services from an unmonetized proliferation through the company’s open API.

Limitations
De Weck’s finding that evolution in complex systems occurs over long timeframes also holds for developer platforms. The field is still nascent, and the chosen examples have been opened for two years. A re-examination, especially regarding the impact of performance of those systems on the overall governance, will provide further insights into the impact of foundational decisions around management and governance of developer platforms (de Weck et al., 2011, p. 65).
A limited perspective on Uber’s internal processes and means for managing risk was published due to the company’s rules to keep such process information confidential.

While the interviews supported the publicly available information around general product development and external governance mechanisms of the developer platform team, the detailed analysis around internal risk and governance mechanisms has been removed.

Further limitations exist for cross-case analysis. These limitations include the possibility of being influenced by vividness during the interview and evaluation process and inadvertently dropping disconfirming patterns in comparison of the cases (Eisenhardt, 1989). The presented research selected consistent dimensions and interviews questions for the companies.

Eisenhardt (1989) adds that reliable theory-building requires four or more cases. In our study, we examined two industry players; hence, this research contributes to theory building but does not constitute new theory.

Further Areas of Research

For academia to build a theory around the rise of external APIs and developer platforms, a broader understanding of the state of this technology across different industrial sectors and types of organizations is necessary. This study examined two cases that are both developed from the ground up on technology companies. Between the two, different approaches to managing and governing their respective developer platforms were identified. New patterns are likely evolving, as additional industries as well as public and private sector firms with a traditionally weaker technology focus are
included in the analysis. The research field touches a multitude of other disciplines, such as theory around product development, open innovation, platforms, organizational structures and governance. Our research focuses on the intersection of those areas, but targeted studies around individual parts can help build a more detailed understanding of dependencies and limitations of those areas for the success and governance of a developer platform within an organization.
# Appendix

## Appendix 1: Interview Questions

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Questions (supporting sub-questions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational Structure</td>
<td><strong>[General Setup]</strong> What is your general setup/ organizational structure?</td>
</tr>
<tr>
<td></td>
<td>- How many employees?</td>
</tr>
<tr>
<td></td>
<td>- Which roles?</td>
</tr>
<tr>
<td></td>
<td>- Where in the organization?</td>
</tr>
<tr>
<td></td>
<td>- Who do you report to?</td>
</tr>
<tr>
<td></td>
<td>- Who, if any, is the executive sponsor?</td>
</tr>
<tr>
<td>Product Development Evaluation Process</td>
<td><strong>[Product Strategy]</strong> How do you evaluate the product roadmap internally? Who is involved, who can veto?</td>
</tr>
<tr>
<td>(Strategic)</td>
<td>- What is the role of the developer platform products?</td>
</tr>
<tr>
<td></td>
<td>- How is the API/developer portal team set up? (roles / number of employees)</td>
</tr>
<tr>
<td></td>
<td>- Who are the main stakeholders at your company outside your team?</td>
</tr>
<tr>
<td></td>
<td>- What is the underlying goal of the investment into your team? Who sets that goal?</td>
</tr>
<tr>
<td></td>
<td>- How do you define vision and roadmap internally? What is the contribution to your firm's business?</td>
</tr>
<tr>
<td></td>
<td>- Who is your user? Who is your customer?</td>
</tr>
</tbody>
</table>

62
| Internal governance process & risk management | [Governance] How do you govern the platform internally? (decision making structures, alignment, formal communications)  
- How do you deal with data ownership at your company?  
- Which governance processes have you set up to reduce competitive risks?  
- Which governance processes have you set up to reduce legal risks?  
- Which governance processes have you set up to reduce engineering risks? |
| --- | --- |
| External governance & partner relations | [Collaboration] How do you manage partner relations? (transparency on roadmap, openness, strict control of terms of service)  
[Philosophy] How open is the platform?  
- How do you assess partner needs/requirements?  
- What is the development principle you are following?  
- How much is your team doing in collaboration with other teams?  
- What is the process around the launch of new developer products?  
- How do you work with partner on an ongoing basis to improve their integrations?  
- How do you manage the expectations? |
| Team steering / KPIs | [General success criteria] What are the key performance indicators for developer platform team? |
Appendix 2: Akamai API License Agreement


This is a License Agreement (the "Agreement") for a certain Application Programming Interface (the "API") owned by Akamai Technologies, Inc. ("Akamai") that is useful in connection with Akamai's service offerings. Reference is made to those certain terms and conditions between Akamai and the Customer currently in place that provide Customer with access to Akamai Services (the “Terms and Conditions”). The individual downloading, installing and/or using the API must be a current, authorized employee of an Akamai Customer (“You”). Your downloading, installing and/or using the API is subject to the confidential information terms found in the Terms and Conditions. You may only have access to the API through proper credentialing and key access provided by Akamai, and any circumvention thereof shall immediately terminate this Agreement and be considered a breach of the terms of your use of the associated Akamai Services under the Terms and Conditions.

Please read this Agreement. By clicking accept/agree, downloading, installing and/or using the API, or any associated documentation, specifications or modified versions of the API (whether customized by Akamai or you), you hereby agree that you are authorized by Customer to bind Customer to this Agreement and to access the API under the Terms and Conditions and these terms. If you do not agree to the terms of this Agreement, or are not a Customer of Akamai under current Terms and Conditions, the API cannot be downloaded, installed or used with your environment or otherwise used for any purpose.

The API may consist of any combination of software code, documentation and/or specifications. Akamai either owns or has the right to provide you the API, portions of which may be copyrighted.

1. Limited License. You have a non-exclusive and non-transferable right and license to use the API. Partners or resellers of Akamai Services under a currently valid reseller agreement with Akamai shall have the additional license to extend their implementation of the API to customers who purchase Akamai Services indirectly through them. The API shall only be used in connection with your authorized use of Akamai’s services, and for no other purpose. Akamai shall have the right to modify the API at any time within its discretion. Updated versions of the API may be made available. You will ensure that anyone who obtains and uses the API does so only in compliance with the terms of this
Agreement. No right to sublicense is granted, and Akamai may terminate this license at any time upon notice to you. All other rights in and to the API are hereby reserved.

2. Restrictions.

(a) Personally Identifiable Information. You will ensure that your use of the API does not provide Akamai any personally identifiable information (PII), or any other information that can be used to uniquely identify, contact, or locate a single person or can be used with other sources to uniquely identify a single individual. Further, Akamai assumes no obligation and is not responsible to obtain end user consent or approval for integration of the API with your environment, or to obtain any such end user consent or approval to collect data using the API.

(b) Third Party Use, Reverse Engineering or Export. You may not use, copy, modify or distribute the API except as provided in this Agreement. Except as set forth herein, the API may not be used in association with any third party product or service including, without limitation, any third party content delivery network (CDN). Except as permitted by applicable law and this Agreement, neither you, nor your end-users, may decompile, reverse engineer, disassemble, modify, rent, lease, loan, distribute, sublicense, or create unauthorized derivative works from, the API or transmit the API over a network. For purposes of clarity, an authorized derivative work is one you create based on the API for your internal use only in conjunction with the Akamai services. You may not use or otherwise export the API except as authorized by United States law and the laws of the jurisdiction in which the API was obtained. In particular, but without limitation, none of the API may be used or otherwise exported or re-exported (a) into (or to a national or resident of) a United States embargoed country or (b) to anyone on the U.S. Treasury Department’s list of Specially Designated Nationals or the U.S. Department of Commerce’s Table of Denial Orders. By using the API, you represent and warrant that you are not located in, under control of, or a national or resident of any country or on any such list.

You agree that Akamai may monitor or audit you use of the API to confirm your compliance with these restrictions.

3. No Warranty On API. You and your end users use the API and any derivative works you may create based on the API at your own risk. Akamai provides the API to you “AS IS” and without warranty and you hereby indemnify Akamai for your use of the API to create derivative works. You are not entitled to any hard copy documentation, maintenance, support or updates for the API, although Akamai may in its sole discretion provide these items.

Akamai expressly disclaims all warranties related to the API, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular
PURPOSE. AKAMAI DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE API WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE API WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE API WILL BE CORRECTED. FURTHERMORE, AKAMAI DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE API OR RELATED DOCUMENTATION IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY OR OTHERWISE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO PORTIONS OF THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

4. Limitation Of Liability. In no event shall Akamai be liable to you for any damages exceeding any amount paid for the API or the service with which the API is used.

UNDER NO CIRCUMSTANCES, INCLUDING NEGLIGENCE, SHALL AKAMAI BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE, INCLUDING, BUT NOT LIMITED TO, DAMAGES RESULTING FROM ANY LOSS OF DATA CAUSED BY THE API. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THIS LIMITATION MAY NOT APPLY TO YOU.

5. Indemnification. You will indemnify, defend and hold Akamai, its employees, agents, consultants, subsidiaries, partners, affiliates, and licensors harmless against any and all claims, costs, losses, damages, liabilities, judgments and expenses (including reasonable fees of attorneys and other professionals) (collectively, “Claims”) that may arise from or are related to (i) use of the API; or (ii) the development, maintenance, use and contents of any derivative work thereof, including but not limited to any infringement of any third-party proprietary rights. At Akamai’s option, you will assume control of the defense and settlement of any Claim subject to indemnification by you (provided that, in such event, Akamai may at any time thereafter elect to take over control of the defense and settlement of any such Claim, and in any event, you will not settle any such Claim without Akamai’s prior written consent).

6. Government End Users. If you are acquiring the API on behalf of any part of the United States Government, the following provisions apply. The API and accompanying documentation are deemed to be “commercial computer API” and “commercial computer API documentation,” respectively, pursuant to DFAR Section 227.7202 and FAR 12.212(b), as applicable. Any use, modification, reproduction, release, performance, display or disclosure of the API and/or the accompanying documentation by the U.S. Government or any of its agencies shall be governed solely by the terms of this Agreement and shall be prohibited except to the extent expressly permitted by the terms of this Agreement. Any technical data provided that is not covered by the above provisions is deemed to be "technical data-commercial items" pursuant to DFAR
Section 227.7015(a). Any use, modification, reproduction, release, performance, display or disclosure of such technical data shall be governed by the terms of DFAR Section 220.7015(b).

7. Controlling Law and Severability. This Agreement shall be governed by the laws of the United States and those of the Commonwealth of Massachusetts. If for any reason a court of competent jurisdiction finds any provision, or portion thereof, to be unenforceable, the remainder of this Agreement shall continue in full force and effect.

8. Miscellaneous. This Agreement constitutes the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior or contemporaneous understandings regarding such subject matter. No amendment to or modification of this Agreement will be binding unless in writing. Your rights under this Agreement will terminate automatically without notice if you fail to comply with any term(s) of this Agreement. Upon termination, You, Customer and any associated end-users will no longer have the right to download, install, integrate and/or use the API or derivate works thereof in any manner.

Appendix 3: Uber Developer Platform Terms of Use

Our Mission

The mission of the Uber API is to make it easy for developers to unlock the power of our logistical network that runs across cities. Through the Uber Developer Portal, we take pride in providing the tools necessary to create new and amazing experiences that help people all over the world discover and use Uber. And with the Uber API Affiliate Program, you have a way to get rewarded for the valuable products you build when your users sign up for and ride with Uber.

At the same time, we need to ensure that the rights of Uber and our users and Drivers remain protected and respected. As such, we offer access to the Uber API subject to the documentation and API Terms below.

We are looking for partners that align with the following principles:
Be a good citizen on Uber's network:

- Respecting the rights and privacy of our users and Drivers is very important to us. Please do not spam or otherwise mislead our users or Drivers in any way.

- User and Driver cancellations happen infrequently; we'd like to keep it that way. Please do not take actions that intentionally increase cancellation rates.

- We work tirelessly to perfect the user pickup so that users can easily find their vehicles and get moving to their final destination. Please do not misrepresent a user's pickup location or destination.

- Many of our Drivers rely on Uber to make a living. Focus on ways to help improve our platform by directing legitimate traffic to our Drivers.

- Be creative and make Uber an even more magical experience for people. Help them get things done more easily and cost effectively with Uber.

Be a strong, trustworthy partner to Uber. Please do not:

- Compete with Uber or try to drive traffic away from Uber.

- Resell Uber's services.

- Aggregate Uber with competitors.

- Store or aggregate Uber's data, except as expressly permitted by Uber.

- Slander or disparage Uber.

Introduction

Uber Technologies, Inc. and its affiliates are excited to provide development access to the Uber API and our applications, websites and technology platform (“Uber Services”). We are seeking the best in developer partners to grow our collective products and services. However, we need to balance offering an open platform for innovative development with ensuring the protection of Uber, the Uber Services and our users and partners. These Uber API Terms of Use (“API Terms”) apply to you. “You” means you individually or the entity that you represent. If you are entering into these API Terms for an entity, you represent and warrant to us that you have the legal authority to bind that entity to these API Terms. “Uber,” “we,” or “us” means (a) Uber Technologies, Inc. if you are using the Uber API or Uber API Materials in the United States, (b) Uber (China), Ltd.
if you are using the Uber API or Uber API Materials in China, and (c) Uber B.V. if you are using the Uber API or Uber API Materials in any other region.

We encourage you to contact us if you have questions or suggestions about the Uber API or these API Terms.

I. Access & Registration

A. Acceptance

All access to and use of (i) the Uber API, (ii) any data accessed or obtained via the Uber API ("Uber Data"), and (iii) Uber API-related documentation, software, and materials provided or made available to you by Uber (such documentation, software, and materials, collectively, the "Uber API Materials") is subject to and must comply with these API Terms. By accessing or using the Uber API, Uber API Materials, or Uber Data, you acknowledge that you have read, and agree to abide by, these API Terms. Please stop accessing and using the Uber API, Uber API Materials, and Uber Data if you determine that you are unable to comply with the current or any future version of the API Terms. The API Terms will evolve as our business and platform expands and as creative developers dream up new applications for the Uber API, so check back often and make sure that you are familiar with the most current version. Any amendment to the API Terms will be effective upon our posting of such updated terms at this location. Your continued use of the Uber API or Uber API Materials after such posting constitutes your consent to be bound by the API Terms, as amended. Additionally, we may modify or update the Uber API from time to time. While we can provide no guarantee, we will try to ensure that future versions of the Uber API are backwards compatible to at least the previous version to the extent reasonably possible.

We reserve the right to revoke access to Uber API without notice if your use of the Uber API violates any of these API Terms or if we otherwise object to your use of the Uber API.

B. Your Account and Registration

To access the Uber API, you will first need to register as an Uber API Developer. You will also need to register each of your applications that use the Uber API with us. All user and application registration information can be found here. After registering, we will issue you with a Client ID and secret code(s) (called the client_id and client_secret ) (each, a “Code”). Your Client ID and Code is required for all calls to the Uber API by your application. You must keep confidential your Client ID and any Code(s) we issue
you. You must keep your registration information accurate, complete, and current for so long as you use the Uber API. You are responsible for all use that occurs under your Client ID, including any activities by you or your employees contractors or agents. If you believe an unauthorized person has gained access to your Client ID or any Code(s) we issue you, you must notify us as soon as possible.

C. Fees

Access to and use of the Uber API, Uber API Materials, Uber Data, and the Uber Services available in connection with the Uber API are currently provided at no charge. However, we reserve the right to charge for access and/or use of the Uber API, Uber API Materials, Uber Data, and/or Uber Services in the future at our discretion, including, without limitation, rated pricing and/or differentiated pricing for business users. We will provide you with notice in the event we decide to start charging for use.

II. Use Policies

A. Acceptable Use

You may use the Uber API and Uber API Materials in connection with your applications, products or services that are registered with us (your "Services") to make available certain various features and functionality of the Uber Services via the Uber API.

Your use of the Uber API, Uber API Materials, and Uber Data may be subject to certain limitations on access, Uber Data requests, and use as set forth in these API Terms, on the Uber Developer Portal, or as otherwise provided to you. If we believe that you have attempted to exceed or circumvent these limitations, your ability to use the Uber API, Uber API Materials, and Uber Data may be temporarily or permanently blocked. We may monitor your use of the Uber API, Uber API Materials, and Uber Data to improve the Uber API or Uber Services and to ensure compliance with these API Terms.

You agree that Uber may collect certain use data and information related to your use of the Uber API, Uber API Materials, and Uber Data in connection with your Services ("Usage Data"), and that Uber may use such Usage Data for any business purpose, internal or external, including, without limitation, providing enhancements to the Uber API, Uber API Materials, or Uber Services, providing developer of user support, or otherwise.
You may include advertisements in your Services near your visual implementation of the Uber API (e.g., banner ads above or below), but you must ensure clear separation between Uber content and any advertisements. Your advertisements should not be displayed in any manner that suggests approval or endorsement by Uber.

**B. Use Restrictions**

In general, we reserve the sole right to determine whether or not your use of the Uber API, Uber API Materials, or Uber Data is acceptable, and to revoke Uber API access for any Service that we determine is not providing added benefit to Uber users and/or is not in the best interests of Uber or our users.

The following are some, but not all, restrictions applicable to the use of the Uber API, Uber API Materials, and Uber Data:

- You may not use Uber API, Uber API Materials, Uber Data, Client ID, or Code(s) for any purpose other than providing the Service for which you are registered to provide as an Uber API Developer.

- You may not use the Uber API, Uber API Materials, or Uber Data in any manner that is competitive to Uber or the Uber Services, including, without limitation, in connection with any application, website or other product or service that also includes, features, endorses, or otherwise supports in any way a third party that provides services competitive to Uber's products and services, as determined in our sole discretion.

- For products that use an upfront fare, you must clearly disclose the upfront fare to end users prior to initiating a ride request on their behalf. Learn more about upfront fares in our Ride Requests documentation.

- You may not share Client ID or Code(s) with any third party except as permitted by us for the use of your Service. You may not share Uber Data with any third party except as permitted by us for the use of your Service or by an end user who has affirmatively consented to the sharing of data about such end user. Uber Data about an end user in your possession or control must be deleted by you upon such end user's request or upon such end user's termination or cancellation of the Service.

- Your Service must allow the end user of your Service to access such end user's Uber Data that you have collected via the Uber API at the request of such end user. Your Service must provide easily accessible end user support contact information.
- You may not collect, store or aggregate Uber Data in any manner except as permitted by us for the use of your Service. You may not share Uber Data (individually, or in the aggregate) with third parties in any manner.

- You may not include or use the Uber API or Uber API Materials in, or in connection with, any application, website or other product or service that includes content that is disparaging of Uber, libelous or may otherwise be perceived as detrimental or harmful to Uber and its business and reputation, in our sole discretion.

- You may not include or use the Uber API or Uber API Materials in, or in connection with, any application, website or other product or service that includes content that is defamatory, libelous, hateful, violent, obscene, pornographic, unlawful, or otherwise offensive, in our sole discretion.

- You may not use the Uber API or Uber API Materials to distribute any virus, spyware, adware, malware, or other harmful or malicious component.

- You may not use the Uber API or Uber API Materials for any purpose which or might overburden, impair or disrupt the Uber Services or related servers or networks.

- You may not use the Uber API, Uber API Materials, or Uber Data to distribute unsolicited advertising or promotions, or to send messages, make comments, or initiate any other direct communication or contact with Uber users or partners.

- You may not, and may not encourage or authorize others to: (i) remove or alter any proprietary notices or marks on the Uber API, Uber API Materials, or Uber Data; (ii) use or access the Uber API, Uber API Materials, or Uber Data for purposes of monitoring the availability, performance, or functionality of any of Uber's products and services or for any other benchmarking or competitive purposes; (iii) use or access the Uber API, Uber API Materials, or Uber Data to aggregate, cache, or store geographic location information or other user information accessible via the Uber API; (iv) frame, wrap or otherwise reproduce significant portions of the Uber Services; or (v) reverse engineer, reverse assemble, decompile, modify or attempt to discover any source or object code of the Uber API, Uber API Materials, any part of the Uber Services, or Uber Data.

- You may not charge end users in any manner for access to or use of the Uber API, Uber API Materials or any services or functionality included in or related to the Uber API or Uber Services. Without limiting the foregoing, you may not sell, rent, lease, sublicense, redistribute or syndicate access to the Uber API or the Uber API Materials, and you may not charge any kind of service, booking or similar fee in connection with any services made available via the Uber Services.
- You will at all times use the Uber API and Uber API Materials in accordance with all applicable worldwide laws and regulations and the Uber User Terms, and you may not use the Uber API or Uber API Materials to conduct or facilitate, in any way, activity that is in violation of applicable worldwide laws or regulations or the Uber User Terms. You will at all times use the Uber Data in accordance with all applicable worldwide laws and regulations, and you may not use the Uber Data to conduct or facilitate, in any way, activity that is in violation of applicable worldwide laws or regulations.

- You must not impose any terms on users of your Service that are inconsistent with these API Terms or the Uber User Terms.

- You agree to comply with the design guidelines, including, without limitation, any attribution requirement(s), which design guidelines may be updated by us from time to time, and you understand and agree that Uber has the sole discretion to determine whether your attribution(s) are in accordance with the guidelines.

- You must ensure that any Uber Data or data otherwise related to your integration of the Uber API is encrypted and transmitted over a secure, encrypted channel (e.g., HTTPS).

- Your use of certain endpoints and scopes is subject to certain limitations on access as established by Uber from time to time. If you desire to implement an endpoint or scope in a manner that would exceed the limitations on access, please contact us.

In addition to, and without limitation of, the restrictions above, if you are granted access to the Request, All Trips and/or Delivery authorization scope(s), the following additional restrictions apply to your use of the Uber API, Uber API Materials, and Uber Data:

- You may not, and may not encourage or allow any third party to interfere with, hinder, limit, or modify any notices or authorization or consent requests provided by Uber.

- You may not initiate or cancel a request for transportation or delivery services without the consent of the end user associated with the trip or delivery request.

- You may not, and may not encourage or allow any third party to interfere with, hinder, limit, or modify the fare amount, fare multiple (including, without limitation, in the event of a "surge" multiple) or any associated fees.

- You may not modify or misrepresent a user’s pickup location or destination, or a delivery pickup or drop off location, in any way.

- You may not use the Uber API or Uber API Materials in any way that would grant someone other than you or the applicable user the right to see any data related
to that user’s trip or delivery (including, without limitation, any pick-up or drop-off
locations, or any trip or delivery routes) without obtaining the prior express
consent of that user.

- You must not use any data obtained from Uber in connection with these scopes
for any advertising or marketing purposes.

- You must destroy any data obtained from Uber in connection with these scopes
within thirty (30) minutes of receipt.

- You may only send push-notifications, SMS messages, or any other notification
or message to a user when such user is on a trip, if (a) you initiated the trip
through your integration of the Uber API, (b) the message is transactional (i.e.,
non-marketing purposes only), and (c) you have obtained the prior consent of
that user.

- You will conspicuously display the following language in any integration of the
Uber API that provides for collection of an individual’s phone number, next to the
field where the individual’s phone number is collected: “I agree that Uber or a
transportation provider or courier may contact me at this phone number
regarding my request for goods or services.”

In addition to, and without limitation of, the restrictions above, if you are granted access
to any of the Driver authorization scope(s), the following additional restrictions apply to
your use of the Uber API, Uber API Materials, and Uber Data:

- You may not use the Uber API, Uber API Materials, and Uber Data for any
advertising, retargeting or marketing purposes.

- You may not share with any third party any information or data that could
reasonably be used to derive any information about the identity of any of your
end users who have attempted to authenticate (or successfully authenticated)
access to any such Driver scope.

C. User Data & Privacy

We respect the privacy of users and expect you to do the same. The basic rule is this:
collect only what you need to provide your Service; collect, use and store it a secure
manner; and retain it only so long as you need it.

You agree to publish and abide by a privacy policy explaining how you collect, store,
use, and/or transfer any Personal Data (defined below) via your Services. You also
agree to comply with all privacy and data protection laws applicable to you. If you are
located outside of the United States, you must disclose this fact to the user and indicate that by using your Service the user may be allowing or enabling the transfer of Personal Data to a country that may offer less protection with respect to Personal Data.

If your use of the Uber API, Uber API Materials, related Uber Services, or Uber Data requires or will likely result in the provision of Personal Data directly to Uber, you agree to obtain all necessary consents and authorizations from the applicable users to provide such Personal Data to Uber. Uber will treat Personal Data obtained from you through your use of the Uber API in accordance with its posted Privacy Policy.

Unless otherwise required by applicable law or agreement with the applicable user to retain such data, if a user revokes the authorization previously granted for your Services to access to their Uber account, you must ensure that all Personal Data pertaining to that user is deleted from your Services and related networks, systems and servers. If you stop using the Uber API altogether or if your Uber API access is revoked, you must delete all Personal Data in the same way.

For the purposes of these API Terms, "Personal Data" means information that may be used, either alone or together with other information, to identify an individual user, including, without limitation, a user’s name, address, telephone number, username, email address, city and country, geolocation, unique identifiers, picture, or other similar information.

III. Rights & Ownership

A. Use of Uber Marks

The rights granted in these API Terms do not include any general right to use the Uber name or any Uber trademarks, service marks or logos (the "Uber Marks") with respect to your Services. Subject to your continued compliance with these API Terms, we agree that you may use Uber Marks for certain limited purposes related to your Services as described below. These rights apply on a non-exclusive, non-transferable, worldwide, royalty-free basis, without any right to sub-license, and may be revoked by Uber at any time. If Uber updates any Uber Marks that you are using, you agree to update such Uber Marks to reflect the most current versions. You must not use any Uber Marks, or any confusingly similar mark, as the name or part of the name of your Services, or as part of any logo or branding for your Services.

Branding and Logos. We may occasionally provide branding elements and logos featuring certain Uber Marks for use within your Services, as further described in our
design guidelines. You can use these resources as we provide, but you must not make any changes or modifications.

Promoting Your App. You may not use any Uber Marks in any advertising, promotional or marketing materials for your Services, provided that you may use the name "Uber" in a written description of your Service in order to describe the fact that your Service connects to the Uber Services using the Uber API. The first time that the name appears in any piece of text, it should be followed by a superscript "®" indicating a registered trade mark (i.e. "UBER®"). You must not use the name in any way that takes unfair advantage of our goodwill or reputation in the Uber Marks, and you must not make any suggestion, directly or indirectly, that your app is an official Uber app or is endorsed by Uber in any way.

Press and Publicity. You may not issue any press release or other announcement regarding your Services that makes any reference to Uber (except as permitted in the "Promoting Your App" paragraph above) without our prior written consent.

B. Ownership

You agree that Uber and its affiliates retain all worldwide right, title and interest in and to the Uber API, Uber API Materials, Uber Data, Uber Marks and the Uber Services, including, without limitation, all intellectual property rights therein. Any rights not expressly granted herein are withheld.

C. Independent Development

You understand that Uber may currently or in the future develop products and services that may be similar to or compete with your Services. Nothing in these API Terms shall in any way restrict Uber from pursuing any business activities or from entering into any agreement with any other person or company.

D. Feedback

In the event that you choose to provide Uber with feedback, suggestions or comments regarding the Uber API, the Uber API Materials, or your use thereof, you agree that Uber will be free to use, copy, modify, create derivative works, distribute, publicly display, publicly perform, grant sublicenses to, and otherwise exploit in any manner such feedback, suggestions or comments, for any and all purposes, with no obligation of any kind to you.
E. Open Source Uber API Materials

To facilitate your Uber API integration, Uber may make certain SDK(s) and/or libraries available to you under a separate open source license. You agree that any Uber API integration facilitated with such open source SDK(s) and/or libraries remains subject to these API Terms.

F. License to Uber

You hereby grant to Uber a paid-up, royalty-free, non-exclusive, worldwide, transferable, sublicenseable, right and license, under all your intellectual property rights, for Uber’s marketing and promotional purposes to: (a) use, perform, make available, display to the public, reproduce, distribute, and import your integration of the Uber API and (b) use your name, likeness, or brand (which includes all your trademarks, service marks, logos, brand names, or trade names) ("Your Marks") to the extent it is incorporated into your integration of the Uber API. Following the termination of these API Terms and upon written request from you, Uber shall make commercially reasonable efforts, as determined in its sole discretion, to remove references to your integration of the Uber API and any of Your Marks from the Uber website.

IV. Legal Terms

A. Confidentiality

You may be given access to certain information and software relating to the Uber API that is not generally known by the public ("Confidential Information"), which is confidential and proprietary to Uber. You agree to use the Confidential Information only for the purpose of using the Uber API in accordance with these API Terms, and you agree to not disclose any of the Confidential Information to any third party without Uber’s prior written consent. You agree to protect the Confidential Information in the same manner that you would protect your own confidential and proprietary information but in no event using less than a reasonable degree of care.

B. Termination

You may terminate these API Terms at any time by ceasing all use of the Uber API, Uber API Materials, and Uber Data, and deleting all copies of all Uber API Materials and all Uber Data in your possession or control. We may immediately revoke your use of the
Uber API, Uber API Materials, and Uber Data without notice if that use violates any of these API Terms or if we otherwise object to your use of the Uber API. We may also terminate any rights granted hereunder, or may add or remove functionalities or features of the Uber API, at any time and for any reason at our discretion. We will not be liable to you or any third party for any costs or damages as a result of termination of this agreement.

C. Disclaimer

THE UBER API, UBER API MATERIALS, UBER DATA, AND UBER SERVICES ARE PROVIDED "AS IS" AND ON AN "AS-AVAILABLE" BASIS, WITHOUT WARRANTY OF ANY KIND, AND UBER DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTIES OR CONDITIONS ARISING OUT OF COURSE OF DEALING OR USAGE OF TRADE. UBER DOES NOT WARRANT THAT THE UBER API, UBER API MATERIALS, UBER DATA OR UBER SERVICES WILL MEET YOUR REQUIREMENTS OR THAT USE THEREOF WILL BE ERROR-FREE, UNINTERRUPTED, VIRUS-FREE, OR SECURE.

D. Limitation of Liability

IN NO EVENT WILL UBER OR ITS EMPLOYEES, AGENTS, USERS OR PARTNERS BE LIABLE TO YOU FOR ANY SPECIAL, INCIDENTAL, EXEMPLARY, PUNITIVE OR CONSEQUENTIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, LOSS OF USE, DATA, BUSINESS OR PROFITS) ARISING OUT OF OR IN CONNECTION WITH THESE API TERMS OR YOUR USE OF THE UBER API, UBER API MATERIALS, UBER DATA, OR UBER SERVICES, WHETHER SUCH LIABILITY ARISES FROM ANY CLAIM BASED UPON CONTRACT, WARRANTY, TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, AND WHETHER OR NOT YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. THE FOREGOING LIMITATIONS WILL SURVIVE AND APPLY EVEN IF ANY LIMITED REMEDY SPECIFIED IN THIS AGREEMENT IS FOUND TO HAVE FAILED ITS ESSENTIAL PURPOSE. UBER'S AGGREGATE LIABILITY UNDER THESE API TERMS WILL NOT EXCEED THE GREATER OF (i) THE FEES YOU PAID TO UBER FOR USE OF THE UBER API MATERIALS IN THE LAST YEAR OR (ii) ONE HUNDRED DOLLARS ($100).

E. Indemnification


You will indemnify, defend, and hold harmless Uber, its affiliates, and their officers, directors, employees, agents, licensors, users and partners from any and all claims, damages, losses, liabilities, actions, judgments, costs, and expenses (including, without limitation, reasonable attorneys’ fees) brought by a third party arising out of or in connection with: (a) your use of the Uber API, Uber API Materials, Uber Data, or the Uber Marks other than as expressly allowed by this agreement; (b) your breach or alleged breach of any of the terms, conditions and representations under these API Terms; (c) your Services or business; or (d) your gross negligence or willful misconduct. You will control the defense and settlement of any claim subject to indemnification by you hereunder, provided that Uber may at any time elect to take over control of the defense and settlement of any claim. You may not settle or compromise any such claim without Uber’s prior written consent.

F. Miscellaneous

If you are using the Uber API or Uber API Materials in the United States, these API Terms will be governed by the laws of the State of California, without regard to conflicts of law principles, and all claims arising out of or relating to these API Terms will be brought exclusively in the federal or state courts of San Francisco County, California, USA, and you consent to jurisdiction in those courts. If you are using the Uber API or Uber API Materials in China, these API Terms will be governed by the laws of Hong Kong, without regard to conflicts of law principles, and all claims arising out of or relating to these API Terms will be brought exclusively in the courts of Hong Kong, and you consent to jurisdiction in those courts. If you are using the Uber API or Uber API Materials in any other country, these API Terms will be governed by the laws of The Netherlands, without regard to conflicts of law principles, and all claims arising out of or relating to these API Terms will be brought exclusively in the courts of The Netherlands, and you consent to jurisdiction in those courts. These API Terms were drafted in English and the English-language version shall control in the event of a conflict with any translated version. You may not assign any of the rights or obligations under these API Terms, by operation of law or otherwise, without the prior written consent of Uber. Any attempted assignment in violation of this paragraph is void. These API Terms constitute the entire agreement among the parties with respect to the subject matter and supersede and merges all prior proposals, understandings and contemporaneous communications. Any modification to the API Terms by you must be in a writing signed by both you and Uber. These API Terms do not create or imply any partnership, agency or joint venture between the parties. No waiver by Uber of any right under this agreement will be effective unless set forth in a writing authorized by Uber. If any part of these API Terms are determined to be invalid or unenforceable by a court of competent jurisdiction, that provision will be enforced to the maximum extent permissible and the remaining provisions of this agreement will remain in full force and effect.
### Appendix 4: Akamai API Overview


<table>
<thead>
<tr>
<th>API Name</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akamai Cloud Networking</td>
<td>Provision and manage Orion Customers, Client and Customer IPSec Keys.</td>
<td>Beta</td>
</tr>
<tr>
<td>Alerts</td>
<td>Create and manage notifications about changes to your traffic patterns.</td>
<td>GA</td>
</tr>
<tr>
<td>Alerts v1</td>
<td>Access active Akamai Luna Control Center Alerts and their details.</td>
<td>Deprecated</td>
</tr>
<tr>
<td>AnswerX Reputation Knowledge Server</td>
<td>Configure custom DNS behavior based on rules stored in Akamai’s Reputation Knowledge Server (RKS).</td>
<td>Beta</td>
</tr>
<tr>
<td>Billing Center</td>
<td>Access to contract usage data in JSON for accounts in which you have access.</td>
<td>GA</td>
</tr>
<tr>
<td>Billing Center v1</td>
<td>Access to contract usage data in CSV format for accounts in which you have access.</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Case Management</td>
<td>Manages support requests to resolve any issues with your Akamai applications and services.</td>
<td>Beta</td>
</tr>
<tr>
<td>Case Management v1</td>
<td>Resolve questions or issues with your Akamai applications and services.</td>
<td>GA</td>
</tr>
<tr>
<td>Cloudlets</td>
<td>Create and manage Cloudlet policies and policy versions.</td>
<td>GA</td>
</tr>
<tr>
<td>Content Control Utility v3</td>
<td>The Content Control Utility (CCU) API allows you to purge edge content by request using</td>
<td>LA</td>
</tr>
<tr>
<td>Service</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Akamai's new Fast Purge capability. At this time, you can invalidate or delete by URL only.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content Control Utility v2</td>
<td>Purge Edge content by request. <strong>Use this API when writing new applications.</strong></td>
<td>GA</td>
</tr>
<tr>
<td>Content Control Utility REST API</td>
<td>Purge Edge content by request. Provided as reference.</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Contracts</td>
<td>Provides access to Akamai contract identifiers and associated products.</td>
<td>GA</td>
</tr>
<tr>
<td>Diagnostics Tools</td>
<td>Explore problems in DNS host-name mapping (dig) and network routing (mtr) from around the world</td>
<td>GA</td>
</tr>
<tr>
<td>DNS – Zone Management</td>
<td>Manage a primary zone’s DNS records</td>
<td>GA</td>
</tr>
<tr>
<td>DNS – Zone Reporting</td>
<td>Retrieve DNS hit information.</td>
<td>GA</td>
</tr>
<tr>
<td>Edge Redirector</td>
<td>This API is deprecated. Use the Cloudlets API to manage the Edge Redirector Cloudlet.</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Enterprise Threat Protector</td>
<td>Manages policy settings to protect against enterprise security related events and acceptable user policy events.</td>
<td>Beta</td>
</tr>
<tr>
<td>Event Center</td>
<td>See Luna Event Center configuration, reporting and alerts events.</td>
<td>GA</td>
</tr>
<tr>
<td>Front-end Optimization</td>
<td>Provides a way to use FEO functionality programmatically rather than through FEO's Configuration Manager.</td>
<td>GA</td>
</tr>
<tr>
<td>HD Network</td>
<td>Build a custom HD Network Management Interface with Akamai Media Services, Domain, Stream, and Archive Management APIs. A pre-OPEN REST API.</td>
<td>GA</td>
</tr>
<tr>
<td>Service</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Image Manager</td>
<td>Maintain a catalog of source images, organize and transform the images, make them available for delivery by the Akamai Intelligent Platform.</td>
<td>LA</td>
</tr>
<tr>
<td>Invoicing</td>
<td>Allows you to access your Akamai invoices.</td>
<td>GA</td>
</tr>
<tr>
<td>Log Delivery Service</td>
<td>View, update and delete configurations for the Portal log delivery service.</td>
<td>GA</td>
</tr>
<tr>
<td>Media Analytics</td>
<td>Manage report packs, and fetch analytic data for a particular report pack.</td>
<td>GA</td>
</tr>
<tr>
<td>Media Reports</td>
<td>Retrieve usage and quality metrics for the Media Delivery products - Adaptive Media Delivery, Download Delivery, RTMP Media Delivery and Object Delivery.</td>
<td>GA</td>
</tr>
<tr>
<td>Media Services</td>
<td>Build a custom HD Network Management Interface with Akamai Utility, Domain, Stream, and Archive Management APIs. Latest (OPEN) version.</td>
<td>GA</td>
</tr>
<tr>
<td>Net Storage</td>
<td>Manage NetStorage FileStore and ObjectStore files and directories. A pre-OPEN API.</td>
<td>GA</td>
</tr>
<tr>
<td>Network Lists</td>
<td>Manage lists to be used by multiple Akamai products and features.</td>
<td>GA</td>
</tr>
<tr>
<td>Prolexic Analytics</td>
<td>Retrieve data from Prolexic DDoS protection and monitoring services.</td>
<td>Beta</td>
</tr>
<tr>
<td>Property Manager</td>
<td>Manage your web content on the Akamai edge network.</td>
<td>Beta</td>
</tr>
<tr>
<td>PublishECCU</td>
<td>Manage ECCU files for purging Edge content. A SOAP API.</td>
<td>GA</td>
</tr>
<tr>
<td>SaaS Registration</td>
<td>Register SaaS applications and customers, and manage their associations.</td>
<td>GA</td>
</tr>
<tr>
<td>SaaS Rules</td>
<td>This API is deprecated. Use the Cloudlets API to manage SaaS Rules.</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Service</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Secure Provisioning Service</td>
<td>Securely provision certifications and hostnames.</td>
<td>GA</td>
</tr>
<tr>
<td>SecureHD Policy Editor</td>
<td>Configure and apply various security services for your Akamai HD Network media content, including Token Authorization, Content Targeting (GEO Protection), Media Encryption and Player Verification protection.</td>
<td>GA</td>
</tr>
<tr>
<td>Security Monitor</td>
<td>Retrieve real-time data on security events.</td>
<td>Beta</td>
</tr>
<tr>
<td>Service-Level Agreement</td>
<td>Accesses SLA test configurations and reports.</td>
<td>Beta</td>
</tr>
<tr>
<td>Site Shield</td>
<td>Provide a Site Shield interface for consumption by nonbrowser clients.</td>
<td>Beta</td>
</tr>
<tr>
<td>Traffic Management</td>
<td>Manage Traffic Management load balancing configurations.</td>
<td>GA</td>
</tr>
<tr>
<td>Traffic Management Load Feedback</td>
<td>POST load data for a GTM domain and GET the current load state.</td>
<td>Beta</td>
</tr>
<tr>
<td>User Admin</td>
<td>Create and manage Users, Groups, and Roles.</td>
<td>GA</td>
</tr>
<tr>
<td>Visitor Prioritization</td>
<td>This API is deprecated. Use the Cloudlets API to manage the Visitor Prioritization Cloudlet.</td>
<td>Deprecated</td>
</tr>
</tbody>
</table>
References


Tiwana, A. (2014). *Platform ecosystems: aligning architecture, governance, and strategy*. Amsterdam ; Waltham, MA: MK.


