The Challenge of API Management: API Strategies for Decentralized API Landscapes

Erik Wilde
Good API
erik.wilde@dret.net

Mike Amundsen
Amundsen Training
mca@amundsen.com

ABSTRACT

The Web as the world's largest information system has largely settled on a solid foundation of HTTP-based connectivity, and the representation of *User Interface (UI)* information resources through a mix of HTML and scripting. In contrast, the similarly rapidly evolving "Web of Services" is still based on a more diverse and more quickly evolving set of approaches and technologies. This can make architectural decisions harder when it comes to choosing on how to expose information and services through an *Application Programming Interface (API)*. This challenge becomes even more pronounced when organizations are faced with developing strategies for managing constantly growing and evolving API landscapes.

This tutorial takes participants through two different journeys. *The first one* is a journey discussing API styles and API technologies, comparing and contrasting them as a way to highlight the fact that there is no such thing as the one best choice. The goal of this first journey is to provide an overview of how APIs are used nowadays in research and in industry. *The second journey* discusses the question of how to define an API strategy, which focuses both on helping teams to make effective choices about APIs in a given context, and on how to manage that context over time when large organizations nowadays have thousands of APIs, which will continue to evolve constantly.

KEYWORDS

API; SOA; Web Services; Open Data

ACM Reference Format:
Erik Wilde and Mike Amundsen. 2019. The Challenge of API Management: API Strategies for Decentralized API Landscapes. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW'19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3308560.3320089

1 TUTORIAL OVERVIEW

APIs are becoming an increasingly important part of the Web. In the same way as the network effect helps the human-facing Web to grow, this is true for the machine-facing Web. There is an ever-increasing set of services available, originating in global and explosively growing scenarios such as connected mobility, smart cities, or the Web of things. On the other hand, developers can build new and innovative applications all the time by connecting with these services, and aggregating them into new services, or into human-facing apps on the Web and other platforms. Even the trend towards native apps on mobile devices wouldn't be possible without the connectivity provided by the Web. The Web simply is much bigger than just the part directly interfacing with humans.

Digital Transformation and similar trends are encouraging organizations to increasingly focus on the "digital shadows" that they are producing: First and foremost, any capability that is being developed must be machine-facing, so that it can become part of the "capability toolbox" of that organization. With this approach, it becomes easier to use capabilities in initially unforeseen ways, because they can be easily accessed and utilized in new (digital) scenarios. On the technical level, this is mostly done through the use of Application Programming Interface (API) approaches, where every single capability in an organization must expose such an API, and ideally is even conceived and designed with that perspective in mind (this often is called an "API-First" approach).

The space of API approaches and technologies is diverse and constantly evolving. One of the reasons for that is that there is no unified global consumption platform as there is for UI Web with the browser. For this reason, the consumption side of APIs is more fragmented, and deciding on how to design APIs is not an easy task. As with any design task, everything starts with understanding the value that a product provides to a consumer, but that becomes harder when the variety of consumers becomes larger, and thus the impact of design decisions on their ease of consumption is harder to assess

- *Part 1: Understanding API Styles and Technologies:* The first part of this tutorial takes participants through the various approaches that API design can be based on nowadays. Both presenters have full-time jobs focusing on API technologies, design, and management, and thus deliver this kind of overview to various audiences on a regular basis. For the WWW2019 tutorial, the plan is to put a special emphasis on how API design for the "Open Web" looks like. In the same way as "Linked Data" and the "Web of Data" look at best practices for opening up data to an open set of consumers, the same can be done for the API space, and this will be covered as part of the first half of the tutorial.
  The UI Web thrives because it is both open and organic: providers have a lot of freedom in how they provide information and services, and in the end it is the "open market of consumers" that decides which approaches and directions ultimately gain popularity. A very similar dynamic happens in the API space: There should be a lot of autonomy for providers to decide on what they use as their design foundations, and how they implement their services. This allows

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3320089

providers to make decisions that are best for the consumers that they design their services for.

But while being open and organic is essential for any system at Web scale to scale over size and time, it also helps that certain design options and decisions are not revisited from the ground up every single time. Instead, it can help a lot if there are communities that establish certain patterns, so that problems can be solved more effectively. On the UI Web, one example where this happens is the space of server-side and client-side Web app frameworks. These often have their periods of popularity, and they represent certain design approaches that were popular at that time. These frameworks come and go, but the Web itself transcends these frameworks, they were simply useful tools at a certain point in time.

- *Part 2: Supporting Continuous Change in the API Landscape:* The second part of the tutorial looks at how the individual design decisions and practices introduced in the first half can scale to scenarios with many APIs. Many organizations are facing the challenge of how to enable and empower their development teams, without centralizing too many of the design and implementation decisions. Organizations want to develop and support an "API culture" that allows them to understand and gently steer the way how APIs are designed and developed, so that ideally the combined value of all APIs and their design and development investments is maximized.

  We introduce the concept of "Continuous API Management" [1], which looks at APIs individually as well as at complete API landscapes. Both presenters have lots of experience in this space with regular presentations at API-focused conferences as well as with training and consulting for large organizations. The goals is to give participants a complete overview of the challenges of API management, both at the individual and at the landscape level.

REFERENCES

[1] Mehdi Medjaoui, Erik Wilde, Ronnie Mitra, and Mike Amundsen. 2018. *Continuous API Management: Making the Right Decisions in an Evolving Landscape.* O'Reilly & Associates, Sebastopol, California.