Identification of API Management Patterns From an API Provider Perspective

Andre Landgraf, 08.02.21, Master's thesis final presentation

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

- Motivation
- Research questions
- Research approach
- Data collection
- Results
- Conclusion

Motivation

Identified research gaps and agendas

Platform and boundary resource literature

- New product architectures: technical and social boundary resources including knowledge management [1]
- Effects of long-term decisions: evolution of platforms and their ecosystems have to be researched [2]
- Boundary resource tuning: more focus on boundary resources required [3, 4]
- Importance of knowledge transfer and communication between the platform owner and application developers [5]

Software ecosystem literature and service-orientation research

- Changing factors: software development through co-creation within software ecosystems [6]
- Technological change: service-oriented architecture emerges into API Economy [7]

API management and API Economy research

- Scientific literature about API management is sparse [8]
- API management lacks conventions [9]
- API Economy: established firms face challenges and adaption pace differs between industries [10]

[1] Yoo et al. (2010) | [2] de Reuver et al. (2018) | [3] Henfridsson and Bygstad (2013) | [4] Eaton et al. (2015) | [5] Islind et al. (2016) | [6] Jansen et al. (2009) | [7] Tan et al. (2016)
[8] Mathijssen et al. (2020) | [9] Sohan et al. (2015) | [10] Bondel et al. (2020)

Research questions

Identify recurring API management concerns and document practical solutions from an API provider perspective:

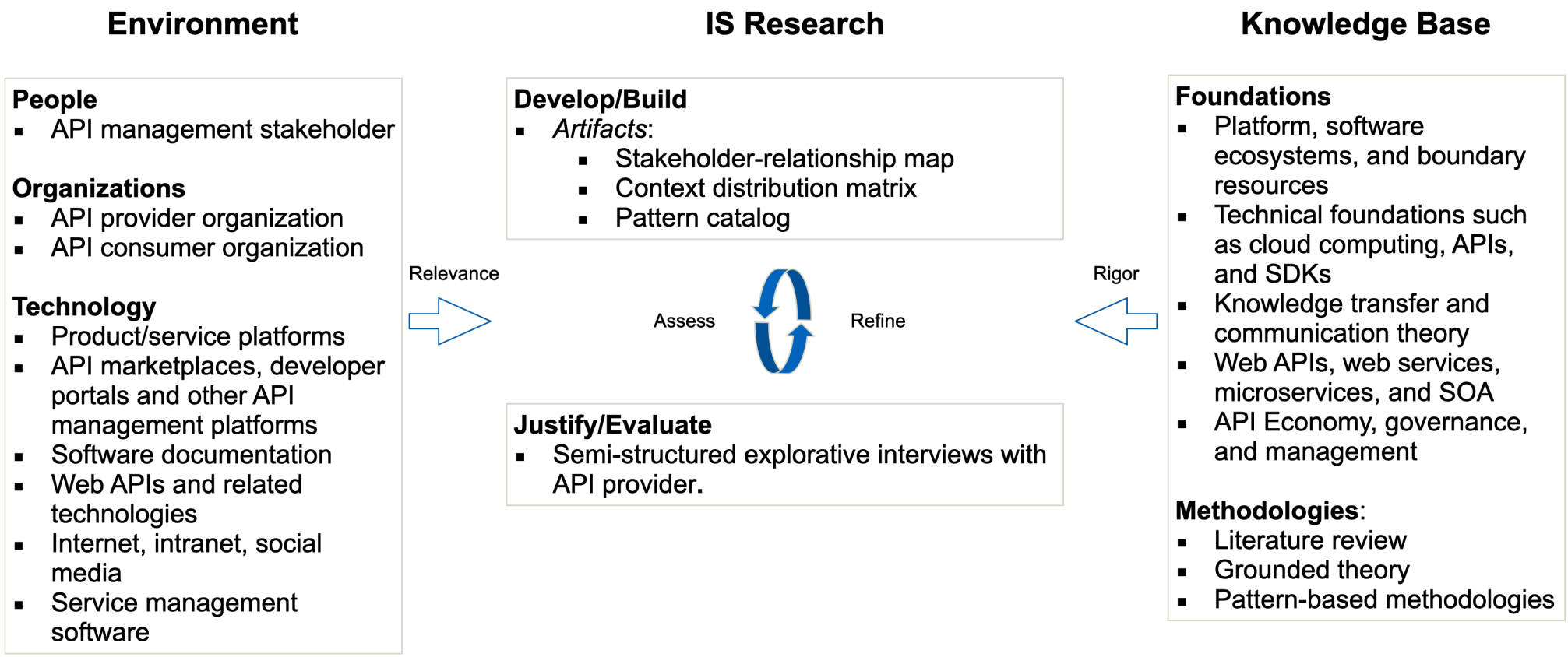
RQ1: What concerns do API providers face in their daily work?

RQ2: What influence factors impact the API management?

RQ3: How do API providers manage concerns and what is the rationale behind the solutions?

Research approach

Design science framework derived from Hevner et al. (2004)



[11] Hevner et al. (2004) | [12] Webster et al. (2002) | [13] Wiesche et al. (2017) | [14] Buckl et al. (2013)

Data collection

Semi-structured interviews with API provider stakeholders

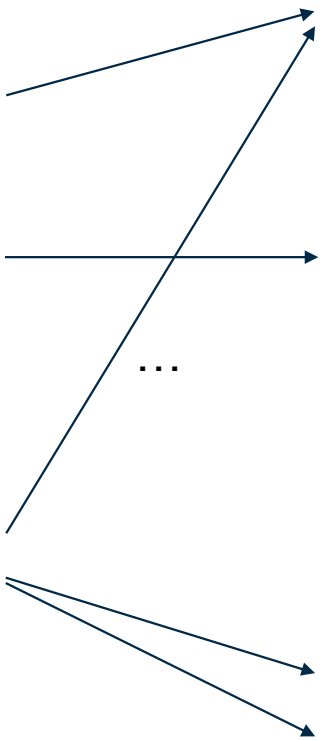
#	Classification	Role	Employees	Duration	Participants
1	Multi-banking startup	Backend Developer	11-50	00:22:52	IV1
2	Industrial manufacturing	Internal Consulting	>100.000	00:44:09	IV2
3	Automotive	Product Owner	>100.000	00:48:49	IV3, IV4
4	Software & IT service provider	Software Architect	1001-5000	00:42:25	IV5
5	IT service subsidiary	Portfolio Manager	1001-5000	00:51:12	IV6
6	Insurance subsidiary	Software Architect	51 - 250	00:59:28	IV7
7	Industrial manufacturing	Technical Lead	>100.000	00:46:34	IV8
8	Industrial manufacturing	Software Architect	>100.000	00:47:03	IV9
9	Financial services	Software Developer	10.001-50.000	00:35:25	IV10
10	Software & IT service provider	Internal Consulting	5001 - 10.000	00:50:49	IV11
11	Software & IT service provider	Integration Architect	11-50	00:56:29	IV12
12	Automotive	Product Owner	>100.000	00:51:48	IV3, IV4
13	Software & IT service provider	Technical Lead, Product Owner	>100.000	00:55:25	IV13, IV14
14	Software & IT service provider	Software Architect	1001-5000	00:50:49	IV5
15	IT service subsidiary	Portfolio Manager	1001-5000	00:31:58	IV6
16	IT service subsidiary	Internal Consulting	1001 - 5000	00:45:44	IV15

Data collection

API platform cases based on a developer portal view

#	Classification	Role	Employees	Duration	Participants
1	Multi-banking startup	Backend Developer	11-50	00:22:52	IV1
2	Industrial manufacturing	Internal Consulting	>100.000	00:44:09	IV2
3	Automotive	Product Owner	>100.000	00:48:49	IV3, IV4
4	Software & IT service provider	Software Architect	1001-5000	00:42:25	IV5
5	IT service subsidiary	Portfolio Manager	1001-5000	00:51:12	IV6
6	Insurance subsidiary	Software Architect	51 - 250	00:59:28	IV7
7	Industrial manufacturing	Technical Lead	>100.000	00:46:34	IV8
8	Industrial manufacturing	Software Architect	>100.000	00:47:03	IV9
9	Financial services	Software Developer	10.001-50.000	00:35:25	IV10
10	Software & IT service provider	Internal Consulting	5001 - 10.000	00:50:49	IV11
11	Software & IT service provider	Integration Architect	11-50	00:56:29	IV12
12	Automotive	Product Owner	>100.000	00:51:48	IV3, IV4
13	Software & IT service provider	Technical Lead, Product Owner	>100.000	00:55:25	IV13, IV14
14	Software & IT service provider	Software Architect	1001-5000	00:50:49	IV5
15	IT service subsidiary	Portfolio Manager	1001-5000	00:31:58	IV6
16	IT service subsidiary	Internal Consulting	1001 - 5000	00:45:44	IV15

API provider interviews

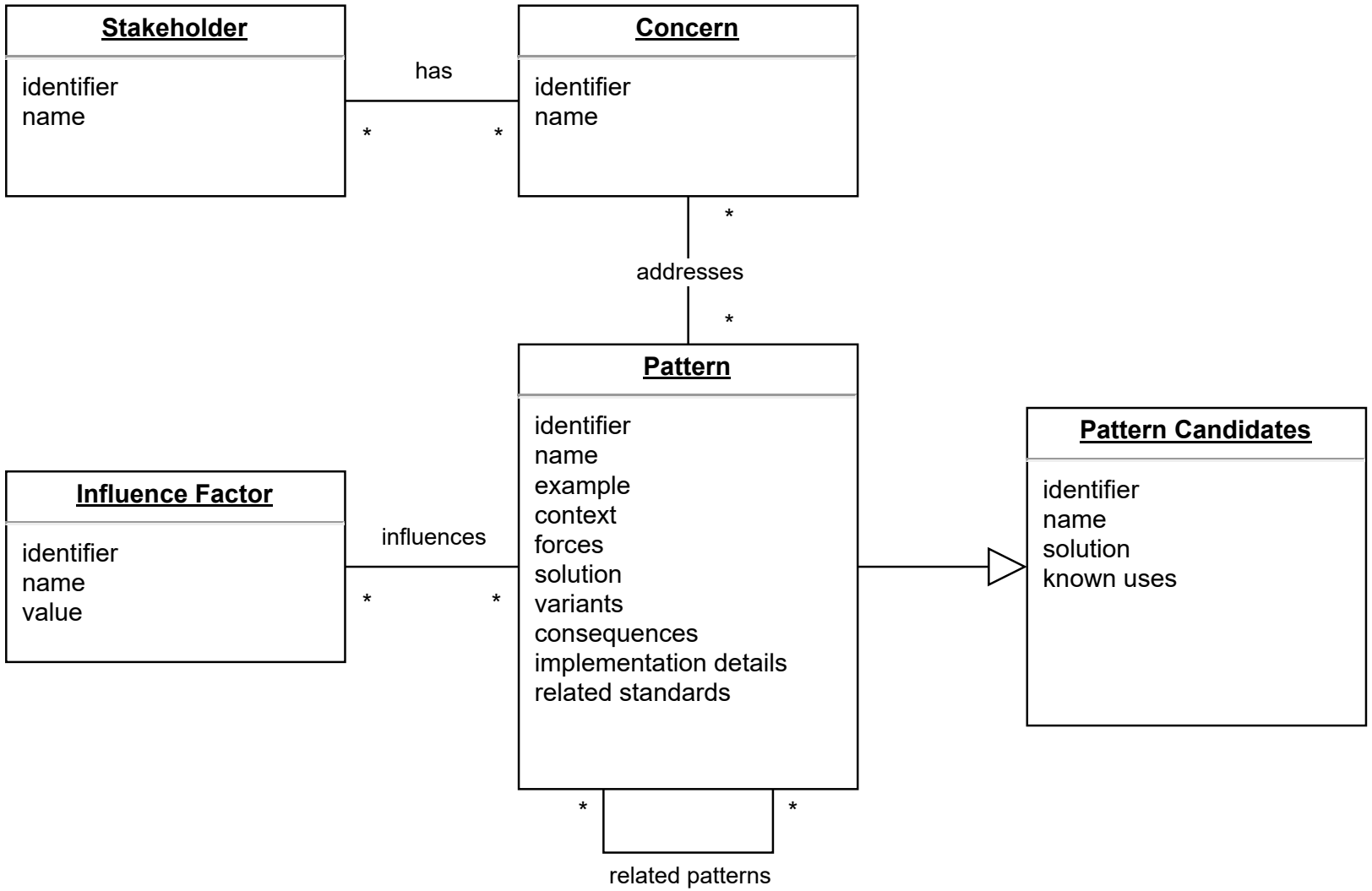


#	# Interview	Architectural Openness	Maturity	Case
1	2	Partner	Pilot	C1
2	3, 12	Public & Partner	Production	C2
3	4, 14	Public	Production	C3
4	4, 14	Partner	Production	C4
5	5, 15, 16	Group	Production	C5
6	6	Group	Pilot	C6
7	7	Private	Development	C7
8	8	Public & Partner	Production	C8
9	9	Partner	Production	C9
10	9	Public & Partner	Production	C10
11	10	Partner	Production	C11
12	11	Public & Partner	Production	C12
13	13	Public & Partner	Production	C13
14	13	Private	Development	C14

API platform cases

Results

Pattern language*

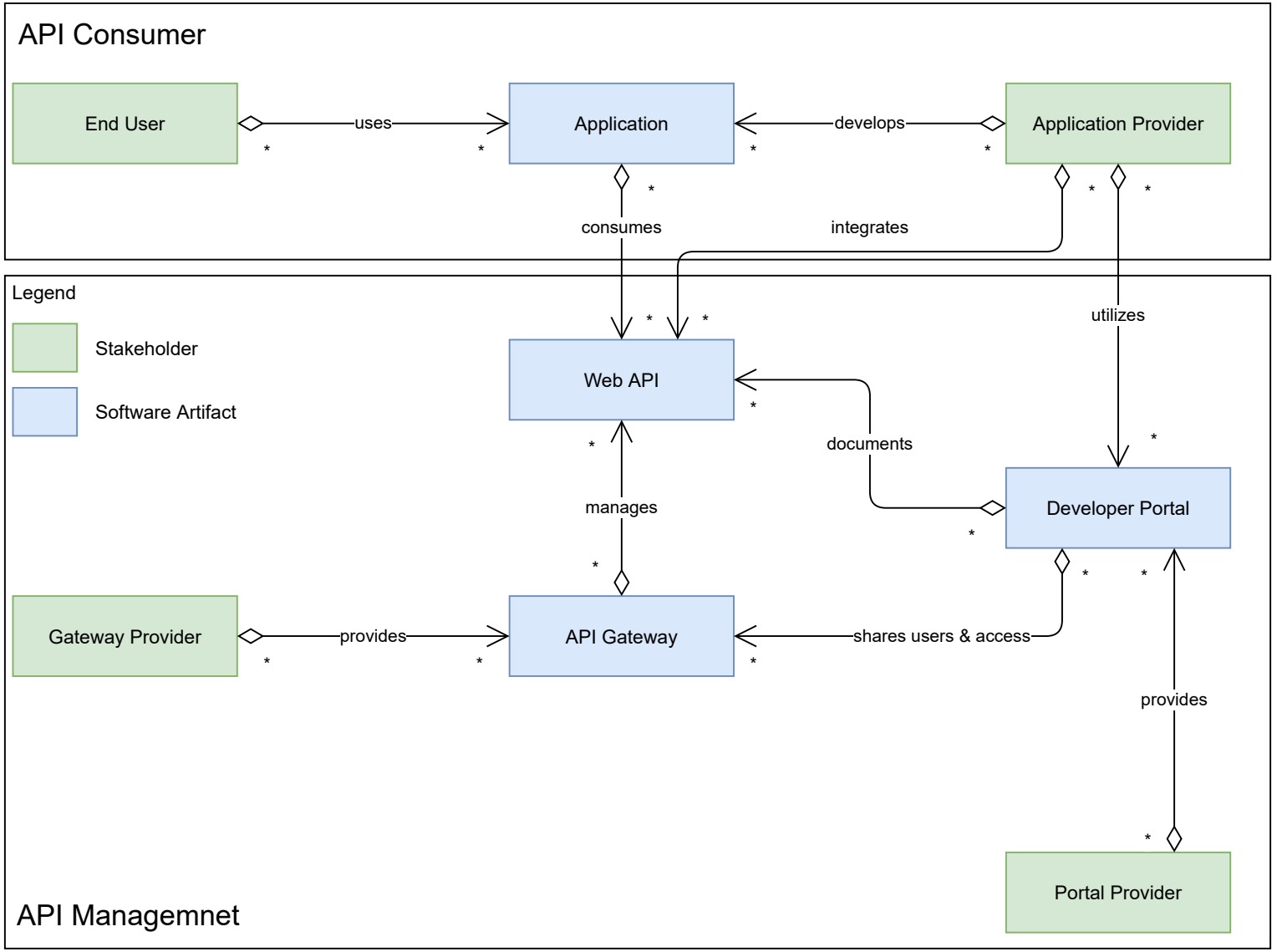


*Based on pattern literature and related pattern languages

[14] Buckl et al. (2013) | [15] Gamma et al. (1994) | [16] Coplien (1994) | [17] Brown et al.(1998) | [18] Lübke et al. (2019) | [19] Zimmermann et al. (2017) | [20] Zimmermann et al. (2020)
[21] Khosroshahi et al. (2015) | [22] Uludağ et al. (2019) | [23] De (2017)

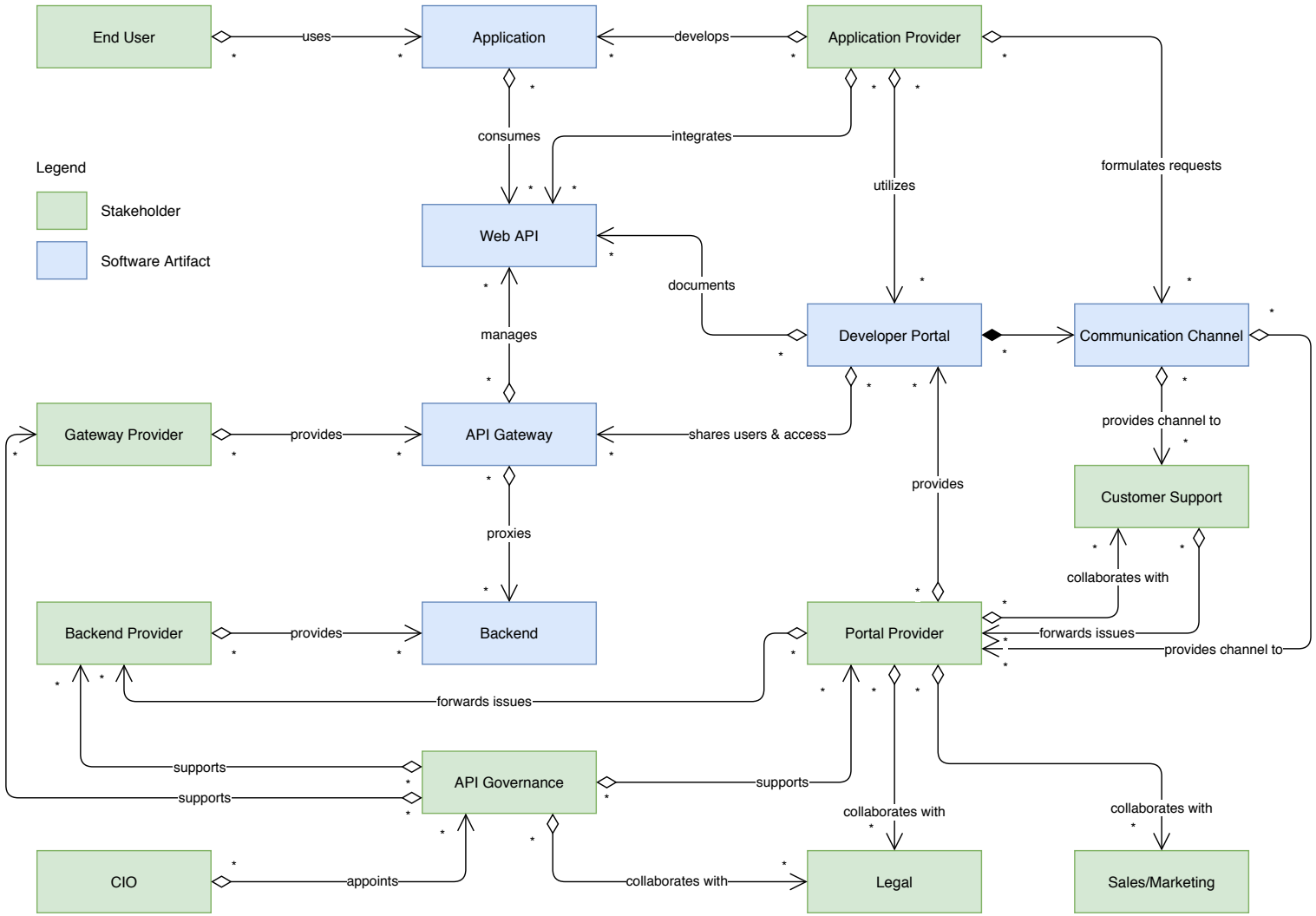
Results

Stakeholder-relationship map



Results

Stakeholder-relationship map



Identified relationships between roles, teams, stakeholders, and IT artifacts of API management

Results

Context distribution matrix*

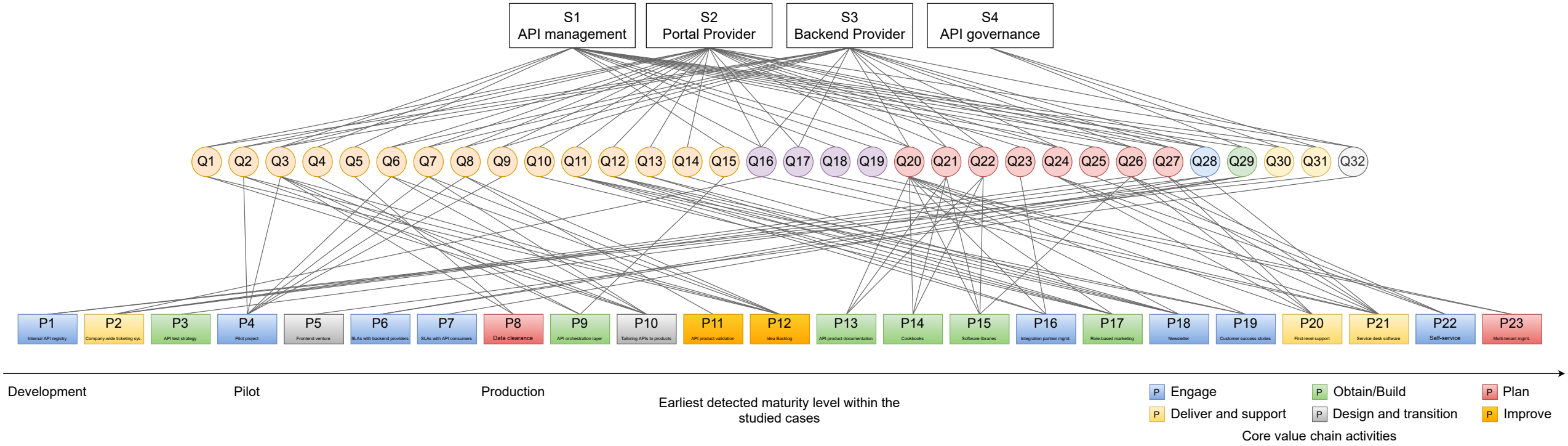
Attribute	Attribute Values			
Architectural Openness	Private [#2, 14%]*	Group [#2, 14%]*	**Partner [#9, 64%]***	Public [#6, 43%]*
Maturity	Development [#2, 14%]	Pilot [#2, 14%]	**Production [#10, 71%]**	
Number of API Consumers	**< 20 [#6, 43%]**	> 20 [#3, 21%]	> 10,000 [#3, 21%]	na [#2, 14%]
Partner Type	**B2B [#12, 86%]***	B2C [#3, 21%]*	B2G [#1, 7%]*	none [#2, 14%]*
Type of Platform	Marketpalce [#2, 14%]	**API Portal [#9, 64%]**	Backend APIs [#2, 14%]	na [#1, 7%]
API Consumer Heterogeneity	Homogenous [#4, 29%]	**Heterogenous [#10, 71%]**		
Monetarization	Free [#3, 21%]*	In Product [#2, 14%]*	**Contractual [#8, 57%]***	Per API call [#6, 43%]*
Initial Driver / Trigger	**Top down [#7, 50%]***	**Bottom up [#7, 50%]***	na [#3, 21%]*	

Context attributes and values with [# of occurrence in cases, percentage of cases] n=14, * denotes multiple counting of cases after Löhe and Legner (2010)

*Derived from encodings and the literature [24] Löhe and Legner (2010)

Results

Pattern catalog - taxonomy and overview



Results

Pattern 12: Idea backlog

Influence Factors

Attribute	Attribute Values			
Architectural Openness	Private	Group	Partner	Public
Maturity	Development	Pilot	Production	
Number of API Consumers	< 20	> 20	> 10,000	na
Type of Platform	Marketplace	Developer Portal	Backend APIs	na
Monetization	Free	In Product	Contractual	Per API Call

Stakeholders

- *Applicants:* S2 Portal provider

Concerns

- Q1 Who will be using the API?
- Q2 Which API should be offered?
- Q3 How to tailor backend services to API that fit the API consumer's needs?
- Q6 How to fit the API to consumers' requirements?
- Q7 How to ensure market-fit?
- Q8 How to validate API offerings?

Solution

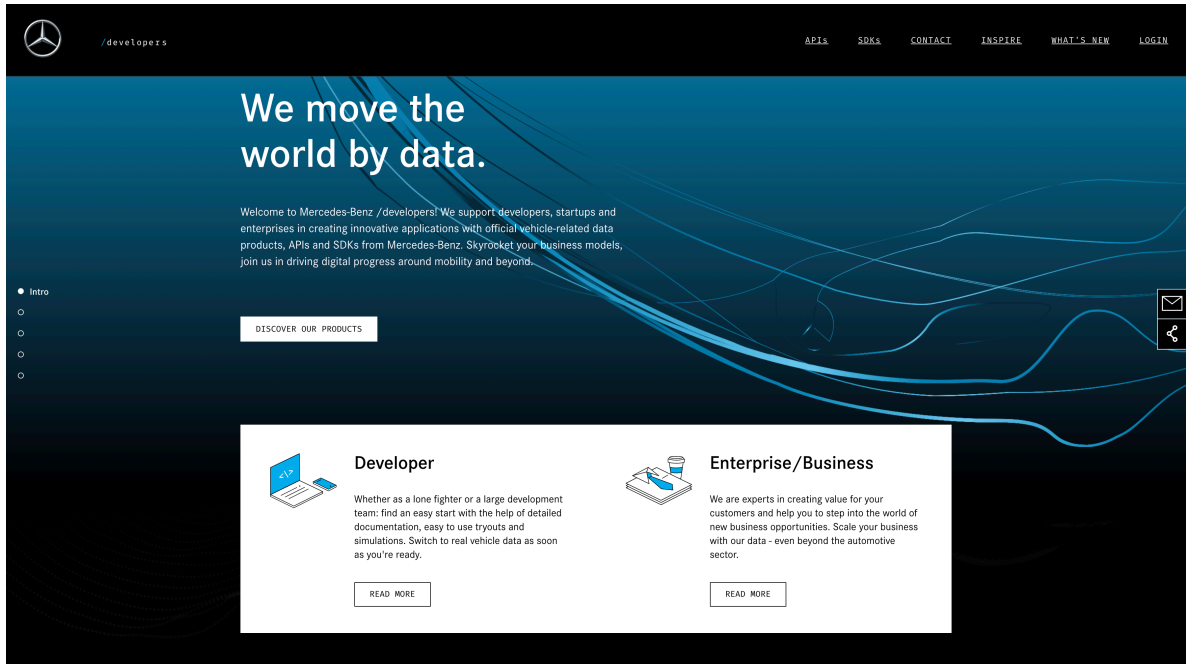
An idea backlog offers a simple and intuitive way to manage incoming feature and change requests. Each request is translated into a ticket within the backlog. It contains information about the requesting (potential) API consumer and a description. Additional fields within the ticket software can be utilized to further enhance the information. Each ticket also provides meta data such as the time and date of the ticket creation that can aid in the analysis of requests.

Known uses: C2, C3, C4

Results

Pattern 17: Role-based marketing

Example: Mercedes-Benz developer portal [A]



Influence Factors

Attribute	Attribute Values			
Architectural Openness	Private	Group	Partner	Public
Maturity	Development	Pilot	Production	
Number of API Consumers	< 20	> 20	> 10,000	na
Type of Platform	Marketplace	Developer Portal	Backend APIs	na
Monetization	Free	In Product	Contractual	Per API Call

Stakeholders

- *Applicants:* S2 Portal provider
- *Potential Collaborator:* Sales and marketing

Concerns

- Q10 How to offer a high-quality user experience for both business and developer roles?
- Q11 How to engage business roles of the API consumer?
- Q12 How to market API offerings to non-technical roles?
- Q13 How to market API offerings to application developers?
- Q20 How to communicate with API consumers?

Solution

Role-based marketing divides marketing material in the developer portal to target different roles of users.

Known uses: Mercedes-Benz, C2, C8, C12

[A] https://developer.mercedes-benz.com/

Conclusion

Realized goals, open goals, and future work

Realized Goals:

- Identification of multiple calls for research and research gaps
- A diverse knowledge base grounded on extensive literature reviews
- 16 conducted interviews with API provider stakeholders
- Creation of three research artifacts
- Identification of 32 concerns and 58 solution approaches

Open Goals and Limitations:

- More follow-up interviews for follow-up questions and validation
- Further evaluation (e.g. pattern workshops [14])
- Comparison of concerns with the literature

Future Work

- Comparison with solution approaches of incumbent software companies
- Investigation of change on SOA based on the emergence of the API Economy
- Longitudinal data required to study long-term effects of decisions [2]
- Further studies about API management [14]

[2] de Reuver et al. (2018) | [8] Mathijssen et al. (2020) | [14] Buckl et al. (2013)

Literature

- [1] Youngjin Yoo, Ola Henfridsson, and Kalle Lyytinen. Research Commentary —The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research. *Information Systems Re- search*, 21(4):724–735, December 2010.
- [2] Mark de Reuver, Carsten Sørensen, and Rahul C. Basole. The Digi- tal Platform: A Research Agenda. *Journal of Information Technology*, 33(2):124–135, June 2018.
- [3] Ola Henfridsson and Bendik Bygstad. The Generative Mechanisms of Digital Infrastructure Evolution. *Management Information Systems Quarterly*, 37(3):896–931, September 2013.
- [4] Ben Eaton, Silvia Elaluf-Calderwood, Carsten Sørensen, and Youngjin Yoo. Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System. *MIS Quarterly*, 39(1):217–243, January 2015.
- [5] Anna Sigridur Islind, Tomas Lindroth, Ulrika Lundh Snis, and Carsten Sørensen. Co-creation and Fine-Tuning of Boundary Re- sources in Small-Scale Platformization. In Ulrika Lundh Snis, edi- tor, *Nordic Contributions in IS Research*, volume 259, pages 149–162. Springer International Publishing, Cham, 2016.
- [6] Slinger Jansen, Anthony Finkelstein, and Sjaak Brinkkemper. A Sense of Community: A Research Agenda for Software Ecosystems. In *2009 31st International Conference on Software Engineering - Com- panion Volume, ICSE 2009*, pages 187–190, June 2009.
- [7] Wei Tan, Yushun Fan, Ahmed Ghoneim, M. Anwar Hossain, and Schahram Dustdar. From the Service-Oriented Architecture to the Web API Economy. *IEEE Internet Computing*, 20(4):64–68, July 2016.
- [8] Max Mathijssen, Michiel Overeem, and Slinger Jansen. Identification of Practices and Capabilities in API Management: A Systematic Literature Review. *arXiv:2006.10481 [cs]*, June 2020.
- [9] S.M. Sohan, Craig Anslow, and Frank Maurer. A Case Study of Web API Evolution. In *2015 IEEE World Congress on Services*, pages 245–252, New York City, NY, USA, June 2015. IEEE.
- [10] Gloria Bondel, Sascha Nägele, Fridolin Koch, and Florian Matthes. Barriers for the Advancement of an API Economy in the German Automotive Industry and Potential Measures to Overcome these Barriers:. In *Proceedings of the 22nd International Conference on Enterprise Information Systems*, pages 727–734, Prague, Czech Republic, 2020. SCITEPRESS - Science and Technology Publications.
- [11] Alan Hevner and Samir Chatterjee. Design Research in Information Systems, volume 22 of Integrated Series in Information Systems. Springer US, Boston, MA, 2010.
- [12] Jane Webster and Richard T. Watson. Analyzing the Past to Pre- pare for the Future: Writing a Literature Review. *MIS Quarterly*, 26(2):xiii–xxiii, 2002.
- [13] Manuel Wiesche, Marlen C. Jurisch, City of Munich, Philip W. Yet- ton, Deaken University, Helmut Krcmar, and Technische Universität München. Grounded Theory Methodology in Information Sys- tems Research. *MIS Quarterly*, 41(3):685–701, March 2017.
- [14] Sabine Buckl, Florian Matthes, Alexander W. Schneider, and Christian M. Schweda. Pattern-Based Design Research – An Iterative Research Method Balancing Rigor and Relevance. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Jan vom Brocke, Riitta Hekkala, Sudha Ram, and Matti Rossi, editors, *De- sign Science at the Intersection of Physical and Virtual Design*, volume 7939, pages 73–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

Literature

- [15] Erich Gamma, Richard Helm, Ralph E. Johnson,and John Vlissides. Design Patterns. Elements of Reusable Object-Oriented Software. Prentice Hall, Reading, Mass, October 1994.
- [16] James O Coplien. A Development Process Generative Pattern Language. page 34, 1994.
- [17] William J Brown, Raphael C Malveau, Hays W McCormick Iii, and Thomas J Mowbray. *Refactoring Software, Architectures, and Projects in Crisis*. John Wiley & Sons, Inc., Canada, 1998.
- [18] Daniel Lübke, Olaf Zimmermann, Cesare Pautasso, Uwe Zdun, and Mirko Stocker. Interface evolution patterns: Balancing compatibility and extensibility across service life cycles. In *Proceedings of the 24th European Conference on Pattern Languages of Programs - EuroPLop '19*, pages 1–24, Irsee, Germany, 2019. ACM Press.
- [19] Olaf Zimmermann, Mirko Stocker, Daniel Lübke, and Uwe Zdun. Interface Representation Patterns: Crafting and Consuming Message-Based Remote APIs. pages 1–36, July 2017.
- [20] Olaf Zimmermann, Mirko Stocker, Daniel Lübke, Cesare Pautasso, and Uwe Zdun. Introduction to Microservice API Patterns (MAP). page 17 pages, 2020.
- [21] Pouya Aleatrati Khosroshahi, Matheus Hauder, Alexander W Schneider, and Dr Florian. Enterprise Architecture Management Pattern Catalog. page 140, November 2015.
- [22] Ömer Uludag˘, Nina-Mareike Harders, and Florian Matthes. Documenting recurring concerns and patterns in large-scale agile development. In *Proceedings of the 24th European Conference on Pattern Languages of Programs - EuroPLop '19*, pages 1–17, Irsee, Germany, 2019. ACM Press.
- [23] Brajesh De. API Management. In Brajesh De, editor, API Management: An Architect's Guide to Developing and Managing APIs for Your Organization, pages 15–28. Apress, Berkeley, CA, 2017.
- [24] Jan Löhe and Christine Legner. SOA adoption in business net- works: Do service-oriented architectures really advance inter- organizational integration? page 16, November 2010.
- [25] Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt. *Patt*ern Oriented Software Architecture: On Patterns and Pattern Lan- guages. John Wiley & Sons, Chichester, April 2007.
- [26] Ajit Kambil. Purposeful abstractions: Thoughts on creating business network models. *Journal of Business Strategy*, 29(1):52–54, January 2008.
- [27] AXELOS Limited and The Stationery Office. *ITIL Foundation: ITIL 4 Edition*. TSO, 4th edition edition, February 2019.

B.Sc.

Andre Landgraf

andre.timo.landgraf@gmail.com

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.17143
Fax +49.89.289.17136

wwwmatthes.in.tum.de

Backup

Timeline

As presented in the kick-off presentation

Activity/Month	August	September	October	November	December	January	February
Kickoff							
Literature review							
Interviews							
Evaluate interviews							
Writing thesis							
Submission							

Data collection

Interview guide

Interview Guide

Introduction

A growing number of companies offer resources through Web APIs instigating the API Economy. Web APIs enable value-adding composition of services that allow new business models. API providers have to manage Web APIs carefully to incorporate changes in the ecosystems while securing internal interests. Key papers have identified a lack of research about Web APIs and stress the importance of longitudinal data. This thesis aims to identify day-to-day issues and actions of API providers through a longitudinal study. The findings will be used to develop pattern candidates that have been discussed with industry experts and API providers.

Terminology

- **API provider**, the team and organization that provides an API
- **API consumer**, the costumer that accesses the capabilities through the API
- **Web API**, APIs that are accessible over the web
- **Public API**, APIs that are accessible to third-party developers outside the organization
- **Private API**, APIs that are accessible inside the organization or to a defined set of partners

Motivation and format

The purpose of this interview is to identify common tasks and challenges of API management and corresponding solution approaches. The interview is planned to be 30 minutes. The interviewee can agree to a set of follow-up interviews to discuss issues, solutions, and activities that emerged since the last meeting. The follow-up interview is meant to be 15-30 minutes.

Terms of confidentiality

The study data will be completely anonymized. We will only connect the following information to the results:

- A short classification of your company
- Your role(s)

This interview will be recorded to be transcribed right after the interview. We will delete the audio/video recording afterwards. Do you agree to recording of this interview? (Yes / No)

Do not hesitate to contact us in case you have any questions or further input.

- Gloria Bondel (gloria.bondel@tum.de)
- Andre Landgraf (andre.timo.landgraf@gmail.com)

Do you have any questions before we start the interview?

Kick-off questions

- How long are you working in IT?
- How old is the API you are working on? Is it released yet?
- Who is involved in the maintenance and development of the API?
- What processes are used for change requests and where do the requirements come from?
- Who is using the API that you are developing?
- How does the communication and collaboration with the API consumers look like?

Current work

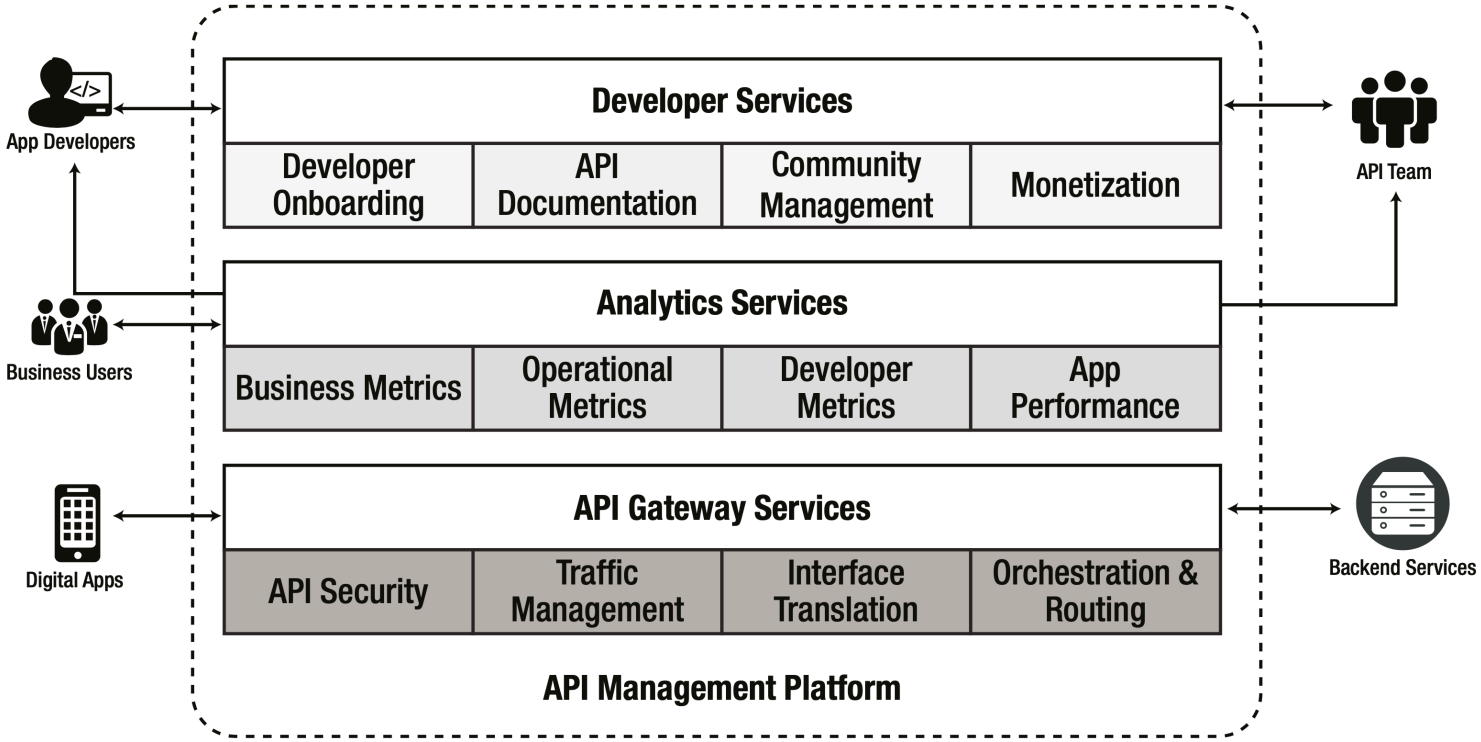
- What are you and your team currently working on?

Follow-up interview questions

- Did you resolve the issue?
- Did it take more or less time than expected? Why do you think that happened?
- Did you communicate the updates with your API consumers? How?
- Were any lessons learned from fixing those issues?

Data collection

API platform hierarchy



API management platforms by De (2017, p. 17)

Developer Portal	Developer Portal
API Gateway	

One gateway can be utilized
by several portals or marketplaces

[23] De (2017)

Results

Context distribution matrix*

Attribute	Attribute Values			
Architectural Openness	Private [#2, 14%]*	Group [#2, 14%]*	Partner [#9, 64%]*	Public [#6, 43%]*
Maturity	Development [#2, 14%]	Pilot [#2, 14%]	Production [#10, 71%]	
Number of API Consumers	< 20 [#6, 43%]	> 20 [#3, 21%]	> 1000 [#3, 21%]	na [#2, 14%]
Partner Type	B2B [#12, 86%]*	B2C [#3, 21%]*	B2G [#1, 7%]*	none [#2, 14%]*
Type of Platform	Marketpalce [#2, 14%]	API Portal [#9, 64%]	Backend APIs [#2, 14%]	na [#1, 7%]
Network Topology	1:1 [#0, 0%]	1:n [#6, 43%]	m:n [#8, 57%]	
Service Granularity	Business Process [#2, 14%]*	Activity & Task [#10, 71%]*	Utility&Entity [#3, 21%]*	na [#2, 14%]*
Offered API Capabilities	Data [#11, 79%]*	Function [#14, 100%]*		
API Consumer Heterogeneity	Homogenous [#4, 29%]	Heterogenous [#10, 71%]		
Monetarization	Free [#3, 21%]*	In Product [#2, 14%]*	Contractual [#8, 57%]*	Per API call [#6, 43%]*
Initial Driver / Trigger	Top down [#7, 50%]*	Bottom up [#7, 50%]*	na [#3, 21%]*	
Number of API calls	Many [#9, 64%]	Few [#6, 43%]		
Value Chain Integration	Vertical [#1, 7%]*	Horizontal [#6, 43%]*	Internal [#2, 14%]*	
Number of API Products	< 20 [#7, 50%]	> 20 [#2, 14%]	na [#5, 36%]	
Onboarding Process	Manual onboarding [#9, 64%]*	Self-service [#6, 43%]*	na [#3, 21%]*	
Network Governance	Focal [#14, 100%]	Polycentric [#0, 0%]		
Networking Target	Efficiency [#5, 36%]*	Innovation [#3, 21%]*	Channel Extension [#6, 43%]*	Venture [#5, 36%]*
Process Output	Virtual [#12, 86%]	Physical [#2, 14%]		
Initial Trigger Motivation	Strategic Pressure [#10, 71%]*	Process Pressure [#0, 0%]*	IS Pressure [#7, 50%]*	
Type of Gateway	Commercial [#8, 57%]	Open source [#2, 14%]	none [#2, 14%]	na [#2, 14%]

Context attributes and values with [# of occurrence in cases, percentage of cases] n=14, * denotes multiple counting of cases after Löhe and Legner (2010)

*Derived from encodings and the literature [24] Löhe and Legner (2010)

Results

Context distribution matrix development

Attributes	From	Derived from	Changed Values
Architectural Openness	Encodings	-	-
Maturity	Encodings	-	-
Number of API Consumers	Löhe and Legner (2010)	Number of Partners	Yes
Partner Type	Löhe and Legner (2010)		Yes
Type of Platform	Encodings	-	-
Network Topology	Löhe and Legner (2010)	-	-
Service Granularity	Löhe and Legner (2010)	-	-
Offered API Capabilities	Löhe and Legner (2010)	Integration Approach	Yes
API Consumer Heterogeneity	Löhe and Legner (2010)	Partner Heterogeneity	-
Monetarization	Encodings	-	-
Initial Driver / Trigger	Encodings	-	-
Number of API calls	Encodings	-	-
Value Chain Integration	Löhe and Legner (2010)	-	-
Number of API Products	Encodings	-	-
Onboarding Process	Encodings	-	-
Network Governance	Löhe and Legner (2010)	-	-
Networking Target	Löhe and Legner (2010)	-	Yes, after Kambil (2008)
Process Output	Löhe and Legner (2010)	-	-
Initial Trigger Motivation	Löhe and Legner (2010)	Pressure	Yes
Type of Gateway	Encodings	-	-

[24] Löhe and Legner (2010) | [26] Kambil (2008)

Results

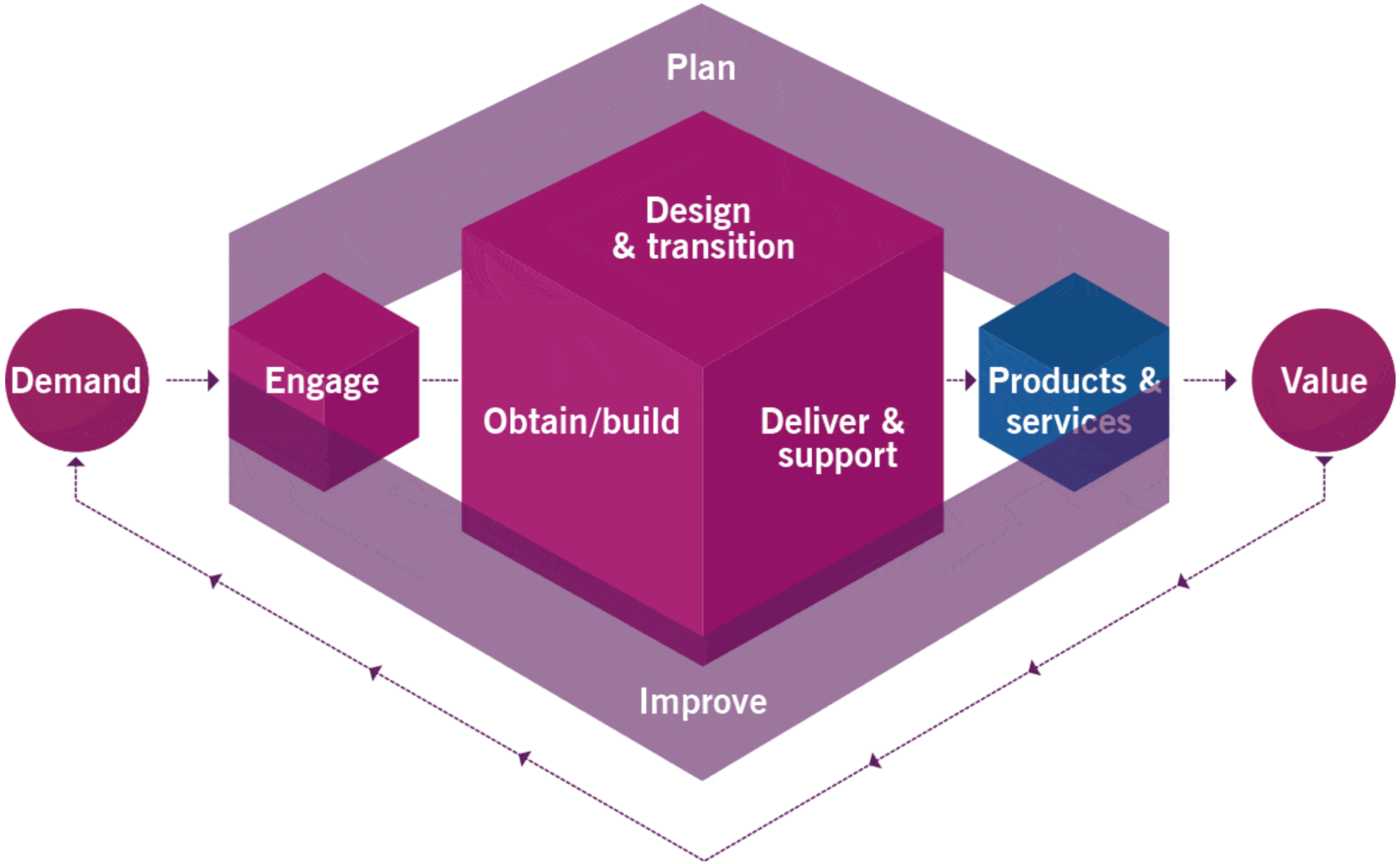
Pattern language development

- A **pattern** is a documented solution for common concerns based on a particular context [14, 15]
- **Stakeholders** are the persons that are involved, affected, or influenced by the domain [22]
- **Concerns** describe the goals, responsibilities, or risks of the stakeholders [22]
- **Context** is utilized to put solution patterns into perspective [25]. We call the most important context attributes for each pattern the **influence factors** after Khosroshahi et al. (2015).
- **Pattern candidates** are validated by the rule of three known uses as established by Coplien (1994).
- **Principles** and **anti-patterns** are not utilized. Patterns provide a fitting framework to document the findings.
- **Pattern form** follows best practices from related pattern languages and the pattern literature [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25].

[14] Buckl et al. (2013) | [15] Gamma et al. (1994) | [16] Coplien (1994) | [17] Brown et al.(1998) | [18] Lübke et al. (2019) | [19] Zimmermann et al. (2017) | [20] Zimmermann et al. (2020)
[21] Khosroshahi et al. (2015) | [22] Uludağ et al. (2019) | [23] De (2017) | [25] Buschmann et al. (2007)

Results

Pattern categories after ITIL (2019)



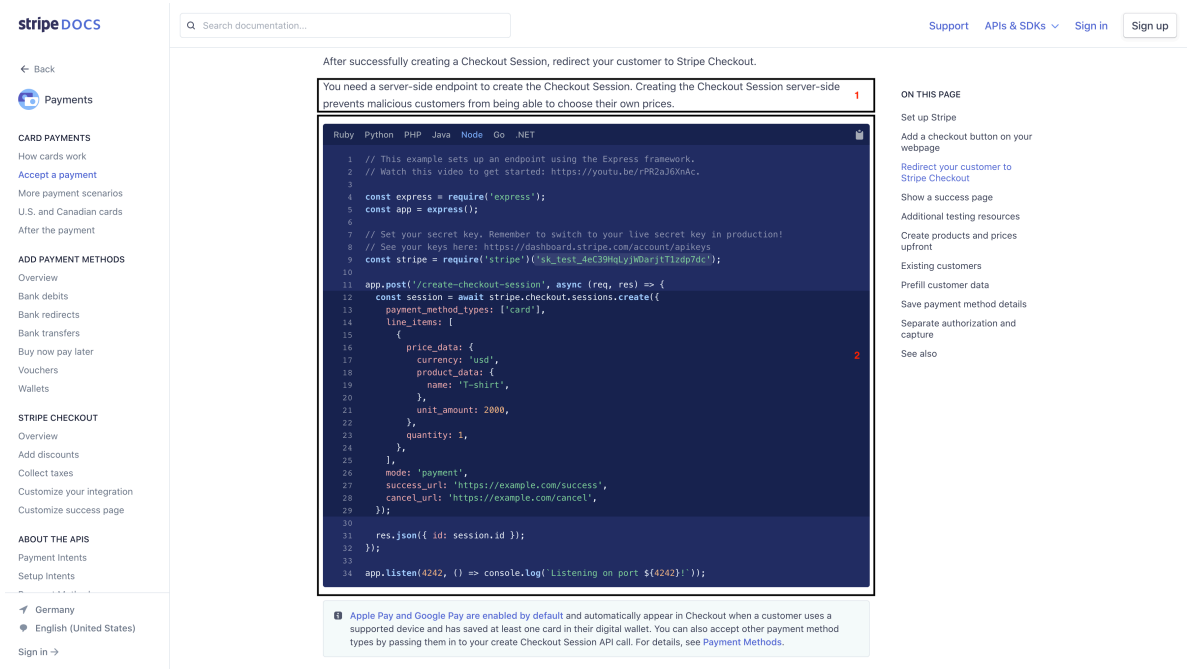
Service value chain activities by ITIL (2019, p. 58)

[27] ITIL (2019)

Results

Pattern 13: Cookbook

Example: Stripe's 'Payments'-product documentation [B]



Influence Factors

Attribute	Attribute Values			
Architectural Openness	Private	Group	Partner	Public
Maturity	Development	Pilot	Production	
Number of API Consumers	< 20	> 20	> 10,000	na
Type of Platform	Marketplace	Developer Portal	Backend APIs	na
API Consumer Heterogeneity	Homogenous	Heterogenous		
Monetization	Free	In Product	Contractual	Per API Call

Stakeholders

- *Applicants:* S2 Portal provider

Concerns

- Q20 How to communicate with API consumers?
- Q21 How to document API products?
- Q22 How to support developers with API integrations?

Solution

Cookbooks are recipe-like, step-by-step integration guides. They describe the API integration from a consumer perspective. Thereby, each user story is documented separately and can be followed in isolation.

Known uses: Stripe, Twilio, C2, C3, C10

[B] https://stripe.com/docs/payments/accept-a-payment

Pattern Candidate 44: Growing FAQ

An FAQ page can help to answer common questions of API consumers. It can further be used to onboard a first level support. A growing FAQ is maintained over time and updated whenever a new common question is identified. It can consist of a public part and a private part. The private part can be used to quickly reuse support responses while the public part can be integrated into the developer portal directly.

Pattern Candidate 46: Support hero

Incoming customer support requests can be disruptive to the current work. One way to handle support requests in an agile way is to create a support hero role. The role assignment rotates every sprint, every week, or bi-weekly between the team members. The support hero has the responsibility to work on all incoming requests. In a Scrum-based environment, the estimated support effort should be considered during sprint planning meetings. Each team of the API management should have its own support hero, e.g. each backend provider, portal provider, and gateway provider team. This ensures that every team within the support chain stays responsive and works on forwarded tickets.