

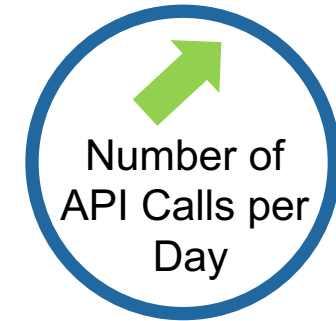
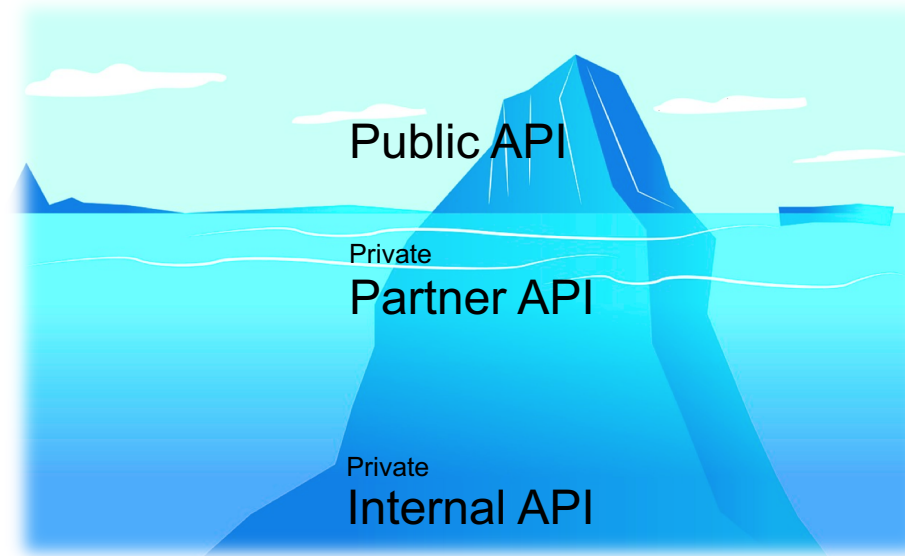
Master's Thesis Final Presentation - Design and Evaluation of a Collaborative Approach for API Lifecycle Management

Duc Huy Bui, 2018-09-10, Garching

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

Outline

1. Motivation
2. Research Questions
3. Research Approach
4. Conception
5. Prototype
6. Evaluation
7. Conclusion



- API Economy
 - API becomes a **product**
 - **Innovation driver** for digital businesses (Cloud, Big Data, IoT, etc.)
 - **Competitive advantages**, e.g. Salesforce (50% revenue), Ebay (60% revenue), Expedia (90% revenue) [1]
- Need for API Management and API Lifecycle Management

Trend: Full Lifecycle API Management

- **Difficulties to align all project members** on the same status of an API
- Unnecessary long and **indirect communication ways**
- **Break in collaboration** between participants e.g. changes of conception and implementation of service or API not trackable
- **No known contact point** for customers who need an API from the company, only through personal network
- **High manual paperwork** and use of a non-optimized process for internal APIs (No existing process for external APIs!)
- Dependencies between frontend and backend developer lead to **longer development cycles for APIs**
- **Bad API usability and missing or bad documentation** of APIs that is not up-to-date
- ...

Need for central solution to guide API consumer and API provider through the API Lifecycle process that is supported by collaboration and acceleration features to efficiently improve shortcomings.

RQ1

How could a holistic approach for an API Lifecycle, including phases, activities, artifacts and roles, look like that is driven by the collaboration of participating stakeholders?

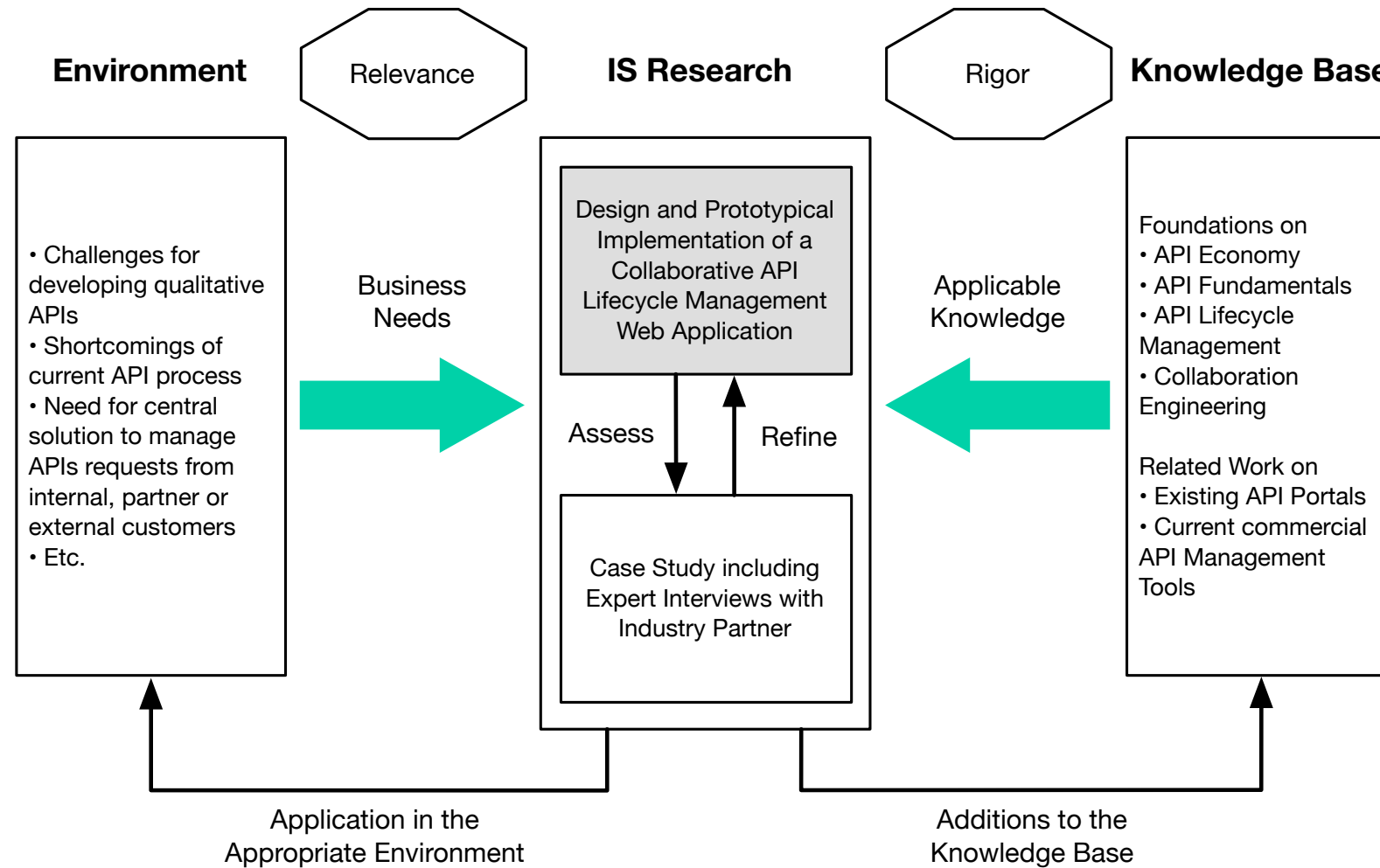
RQ2

How can tools and collaborative features be used to support the API Lifecycle Management?

RQ3

What are the users' experiences of the designed web application prototype solution?

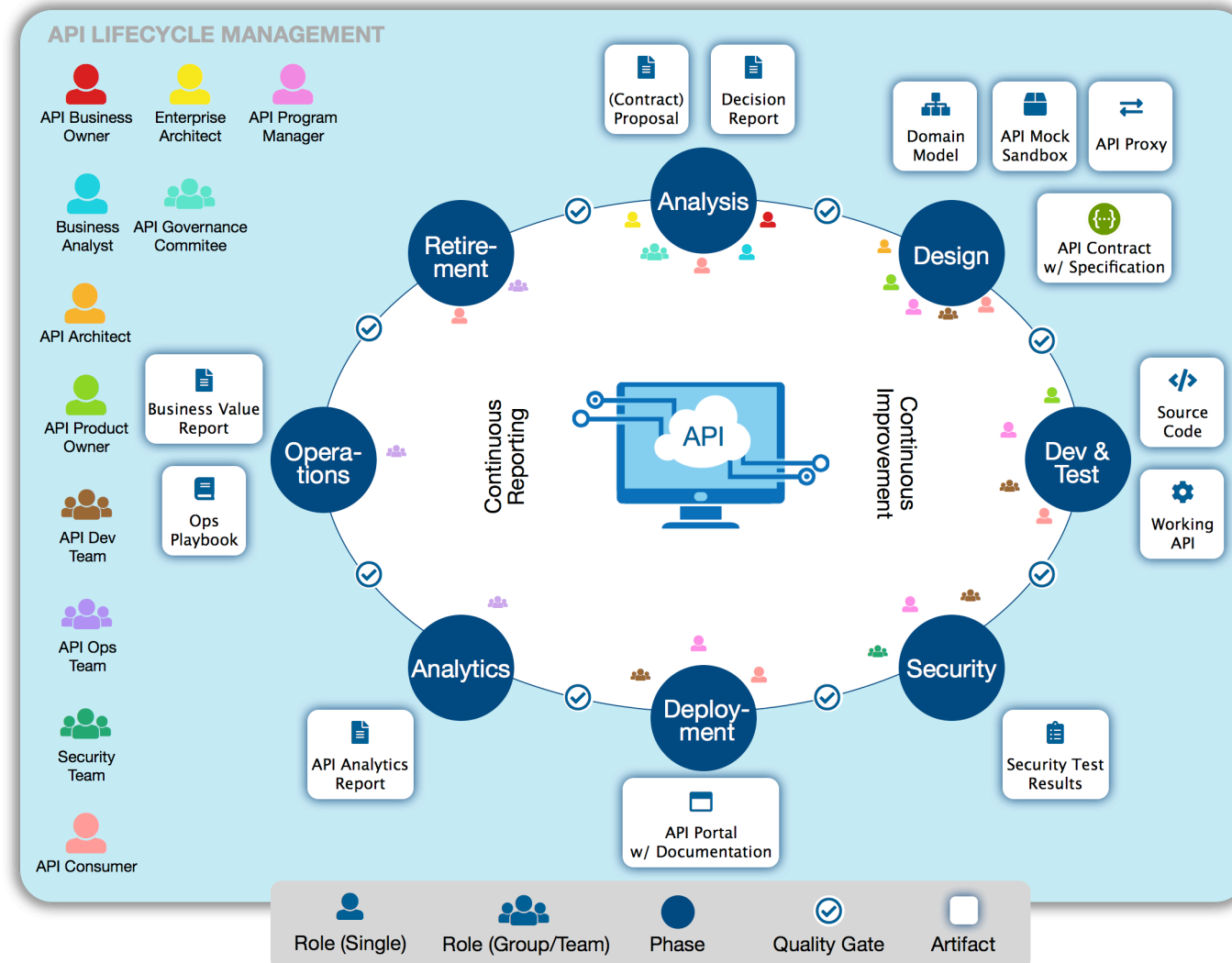
Research Approach



Adapted from Hevner, March, Park, & Ram. (2004). *Design Science in Information Systems Research*. *MIS Quarterly*, 28(1), 75. <https://doi.org/10.2307/25148625>

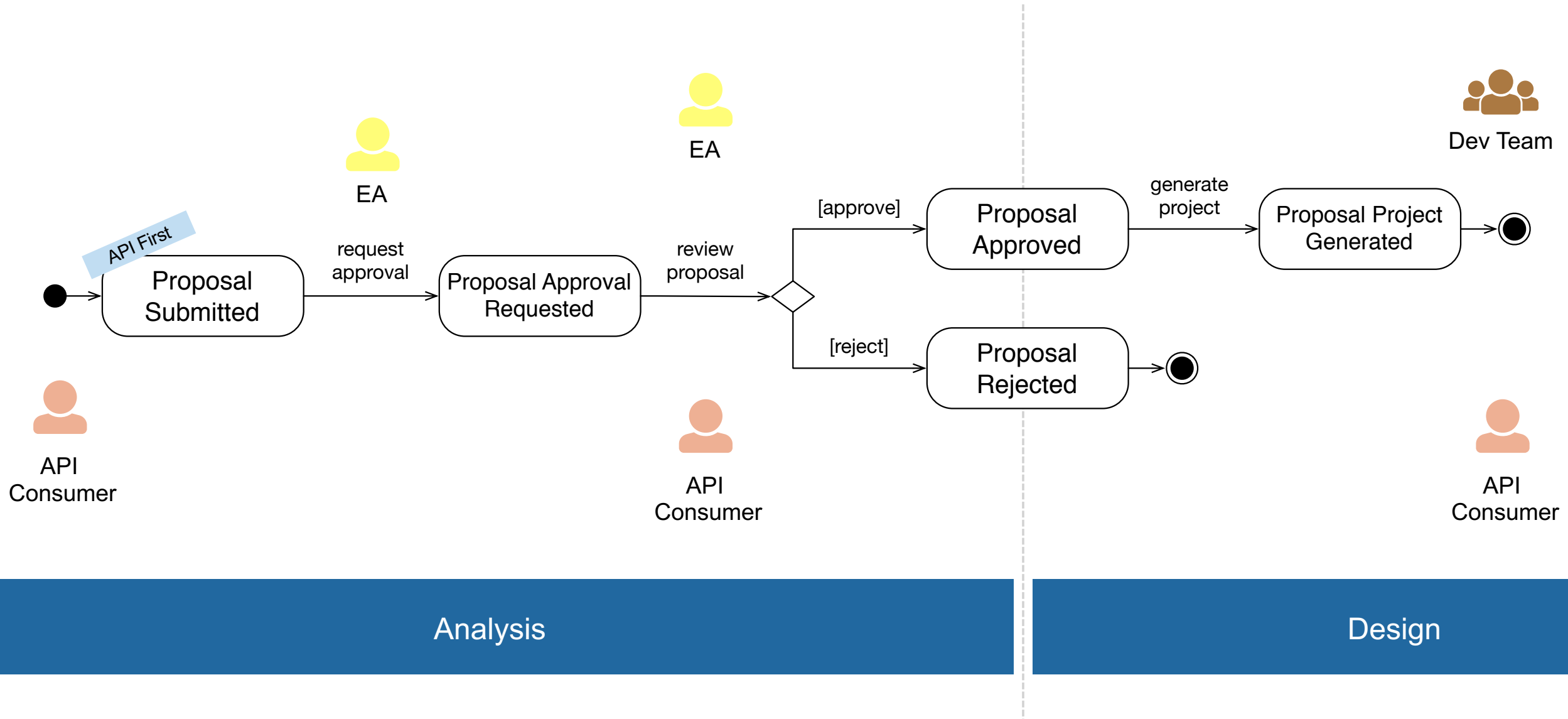
Collaborative API Lifecycle Management

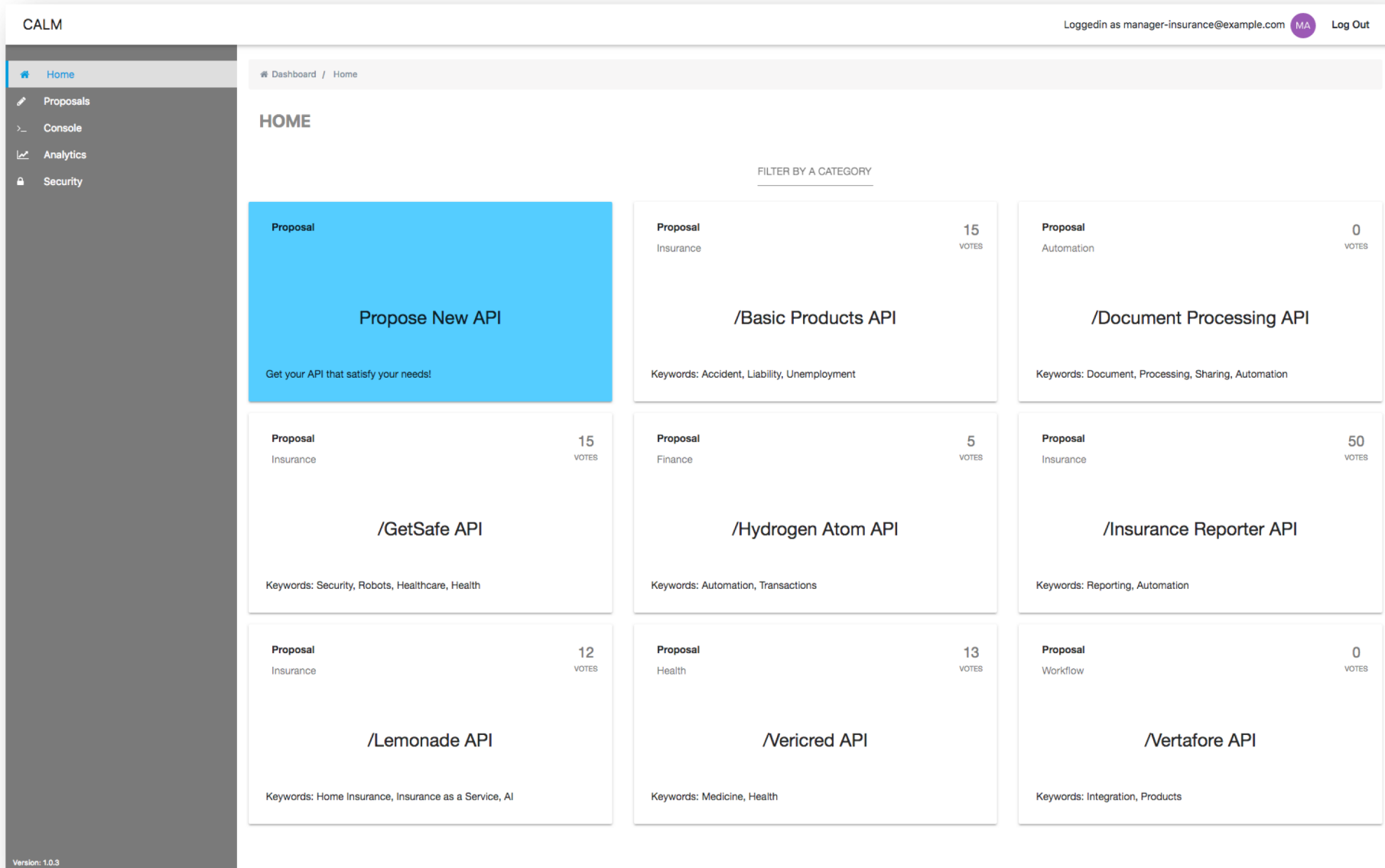
Extract from Conceptual Model



Prototype

States and Activity Flow





The screenshot displays the CALM dashboard interface. At the top left, the text "CALM" is visible. On the right side of the top bar, it says "Logged in as manager-insurance@example.com" with a user profile icon and a "Log Out" button. A sidebar on the left contains navigation links: Home, Proposals, Console, Analytics, and Security. The main content area is titled "HOME" and includes a "FILTER BY A CATEGORY" link. Below this, there is a grid of proposal cards. Each card shows the proposal title, category, and the number of votes. A prominent blue card at the top left is labeled "Propose New API" and contains the text "Get your API that satisfy your needs!".

Proposal Title	Category	Votes	Keywords
Propose New API	-	-	-
/Basic Products API	Insurance	15	Accident, Liability, Unemployment
/Document Processing API	Automation	0	Document, Processing, Sharing, Automation
/GetSafe API	Insurance	15	Security, Robots, Healthcare, Health
/Hydrogen Atom API	Finance	5	Automation, Transactions
/Insurance Reporter API	Insurance	50	Reporting, Automation
/Lemonade API	Insurance	12	Home Insurance, Insurance as a Service, AI
/Vericred API	Health	13	Medicine, Health
/Vertafore API	Workflow	0	Integration, Products

Version: 1.0.3

Goal

Evaluation of the usability and utility of the proposed solution for Collaborative API Lifecycle Management

7 Interview Partners
(2 RA, 4 EA, 1 SWA)

Scenario 1: API Consumer

- Search for existing API proposal
- Create new API proposal with details like API Specification, SLAs, etc.

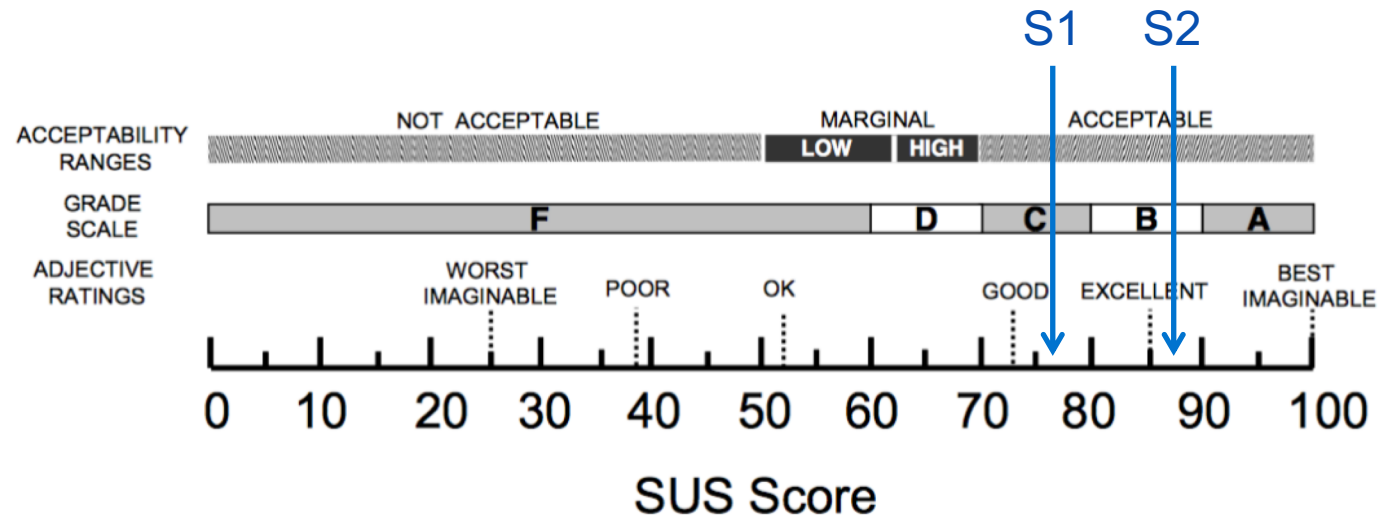
Scenario 2: API Provider

- Review API proposals
- Prepare approved proposals for handover

System Usability Scale (SUS)

4 Open Qualitative Questions

ID	Alias	Score of Scenario 1	Score of Scenario 2
1	RA1	77.5	-
2	RA2	35	-
3	EA1	77.5	87.5
4	EA2	87.5	95
5	SW1	90	77.5
6	EA3	95	92.5
7	EA4	85	87.5
Average		78.21	88.0



Source for SUS Score Scale: Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Studies*, 4(3), 114–123.

Highlights

- **General idea** of a central entry point in form of a web portal
- **Social features** like comments and votes are useful
- **API Specification linter** to pre-check violations helps for unification and structure
- **Automated project generation** (Documentation, Repository, CI/CD Pipeline)
- **Integration of existing tools** like Github

Enhancement Ideas

- Include a **history tracker** to record changes of a proposal
- Rethink the **staging model** and adapt to API Type
- Provide more **context information** and **examples** about constraints beforehand
- Improve **review system** and provide possibilities for iterations
- Stronger focus on **user-centered design** for API Consumer (data model, SDK, etc.)

Prototype provides *Acceleration*?

YES

Prototype helpful for *Collaboration*?

YES

NEUTRAL

Goal

- Prototype to support the Collaborative API Lifecycle Management
- Collaboration features to align users and reach higher product quality
- Acceleration approach to facilitate life-cycle

Solution Approach

- Requirements for conceptual model
- Collaborative API Lifecycle Management model
- Design and prototypical implement of main design artifact
- Evaluation in form of case study with expert interviews

Key Findings

- Prototype is a viable solution
- API Linter provides flexible and simple governance check
- Acceleration through structured form, transparency and automation approach
- Collaboration features helpful, but require active commitment
- API First supported by architects, but doubted by developers

Future Work

- Further DSR iterations and include feedback from 1st iteration
- Evaluation of prototype with other stakeholders from other companies
- Evaluation of conceptual model in other companies
- Increase usability by including Gamification and Machine Learning
- Consideration of further API protocols beside REST

Thank you for your attention! 😊



B.Sc.

Duc Huy Bui

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

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.
Fax +49.89.289.17136

duchuy.bui@tum.de
www.matthes.in.tum.de

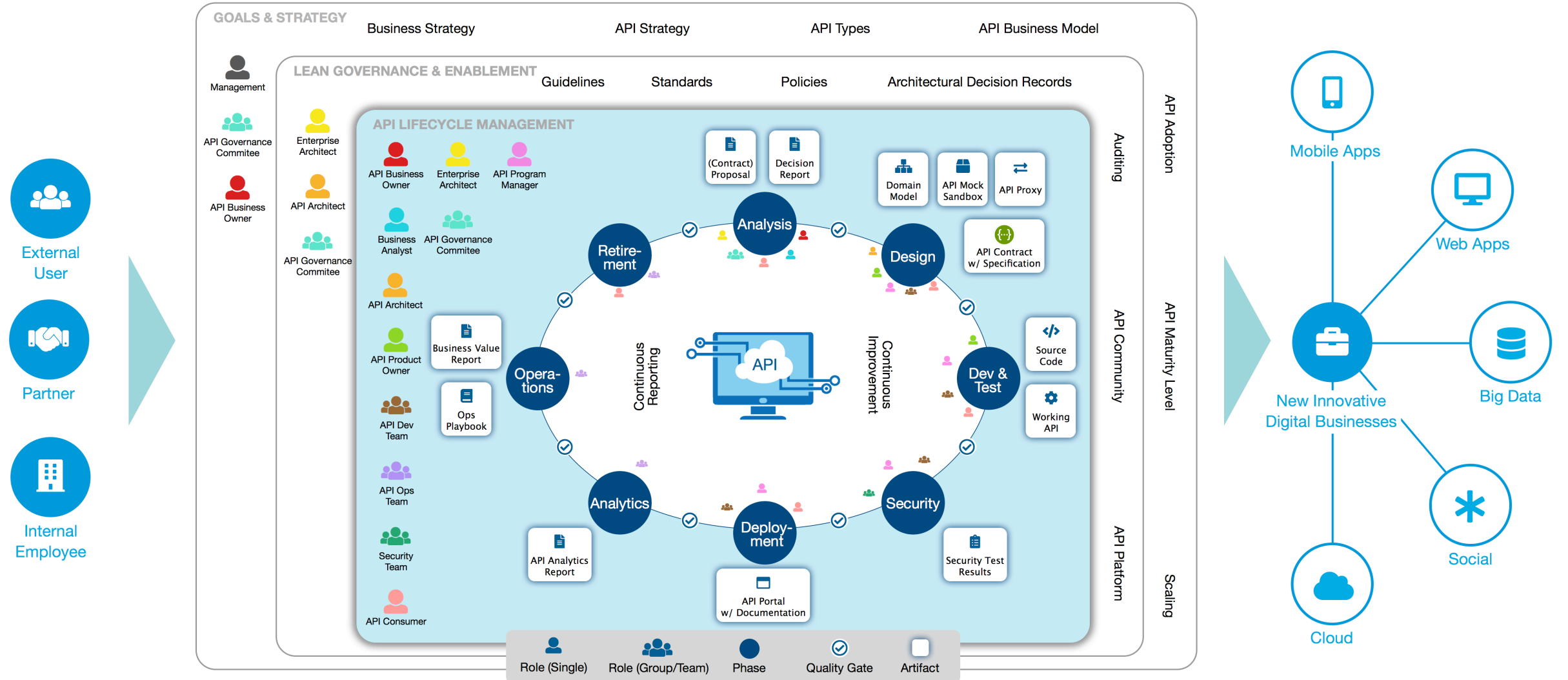


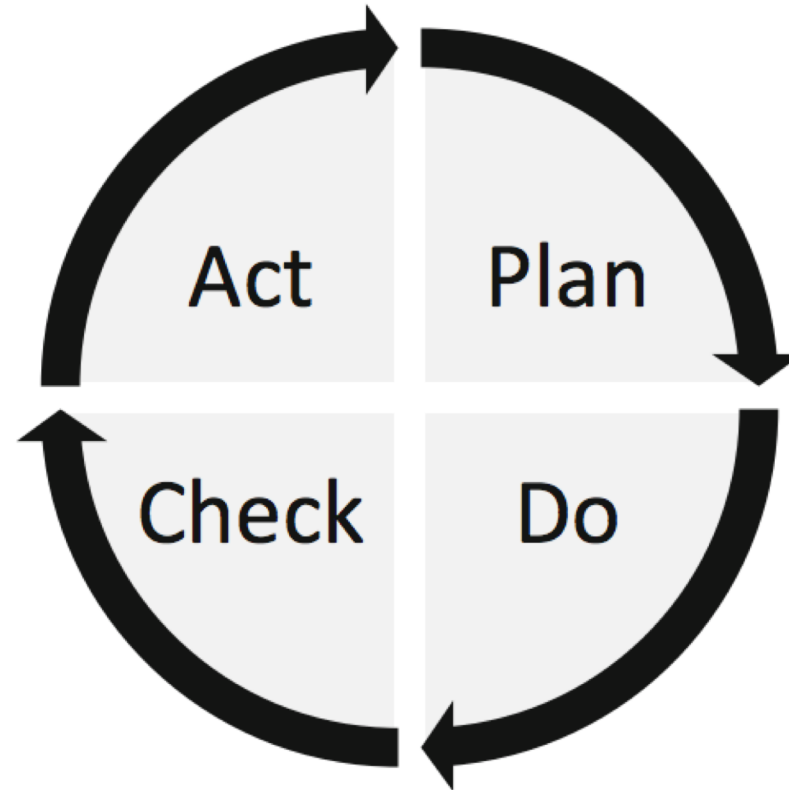
Backup

- Sources for success factors (8)
 - API Management/Lifecycle (industry (2), academic papers (1))
 - Product Development (2)
 - (Enterprise-/IT-) Service Management (2)
 - Agile Software Development (1)
- Structure success factors into categories
 - Business
 - Organizational
 - Process
 - Technical
- Result: [Consolidated table of requirements for API Lifecycle](#)
- Example requirement: “Design for UX/DX” or “Top Management Support”

Collaborative API Lifecycle Management (CALM)

The big picture of Full Lifecycle API Management



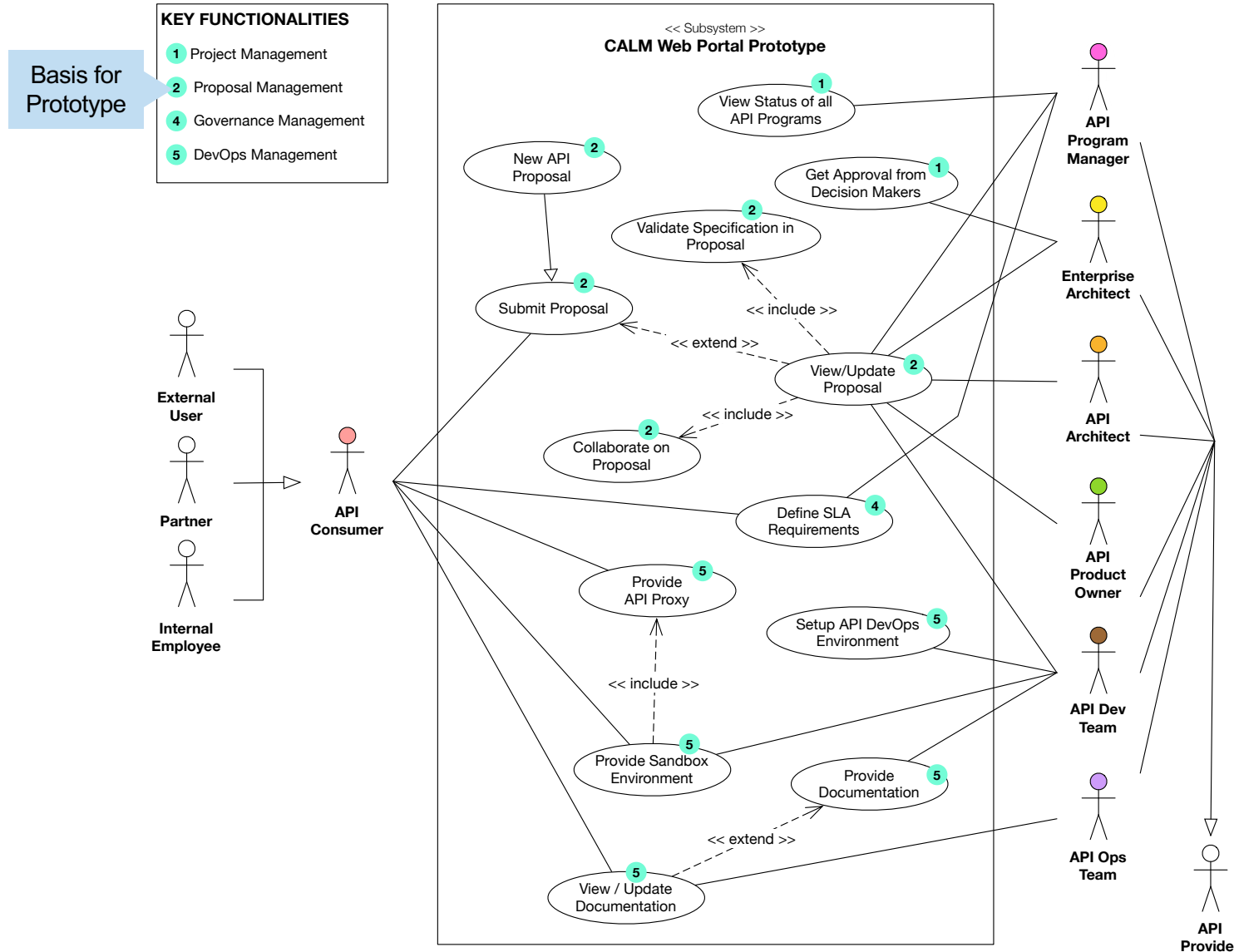


Source: Gemechu, E. D., Sonnemann, G., Remmen, A., Frydendal, J., & Jensen, A. A. (2015). How to Implement Life Cycle Management in Business? In G. Sonnemann & M. Margni (Eds.), *Life Cycle Management* (pp. 35–50). Dordrecht: Springer Netherlands.

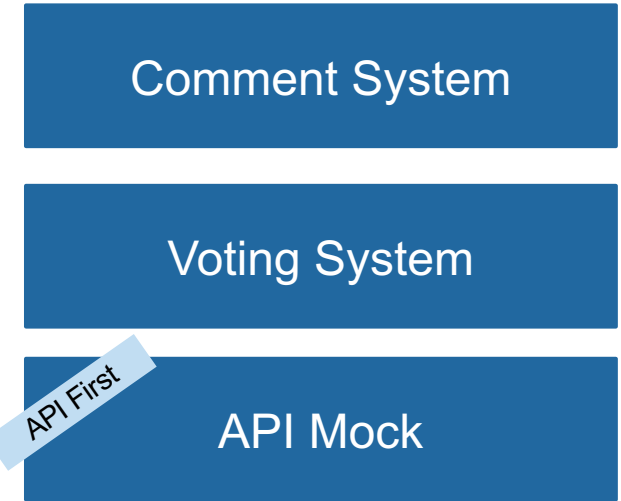
Product Focus (SDLC)	Service Focus (ITSM)
Planning	
Negotiate scope based on function	Negotiate scope based on end-to-end business process
Internally (IT) focused	Customer focused
IT jargon	Business jargon
Requirements Modeling	
“Over the wall” mindset	Stakeholder involvement
Technology insights	Business metrics
Automate function once and move on	Automate service once, reuse service in different ways
Focus on inputs and outputs	Focus on business needs and process
Design	
Capture logic of business function	Model business rules and external relationships
Focus on IT artifact	Focus on end-to-end business services
Construction	
Create a software product	Increase focus on value-added portions of applications
Buy, build or lease	Buy, build, lease and INTEGRATE
Deployment	
Technology driven	Minimum impact on business services
Test technology	Test service environment
Train on technology	Train in business service/process
Support	
Maintain hardware/software/networks	Continual service improvement
Table 1 –Product vs. Service Focus in SDLC	

Life Cycle	ITILV3 (service focus)	Systems Development (product focus)
S	• Service Strategy	• Project Planning
	• Service Design	• Requirements Modeling
T		• Design
		• Implementation (Construction)
A	• Service Transition	• Conversion (Deployment)
G	• Service Operations	• Support
E	• Continual Service Improvement	
Table 2 – A comparison of the SDLC and ITIL		

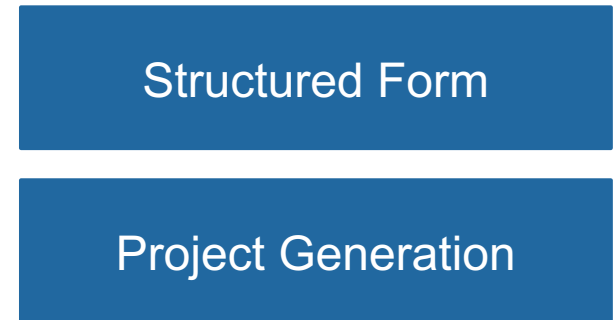
Source: Pollard, C. E., Gupta, D., & Satzinger, J. W. (2009). Integrating SDLC and ITSM to “Servitize” Systems Development. *AMCIS 2009*.



Collaboration Features



Acceleration Features



Prototype

RBAC Model

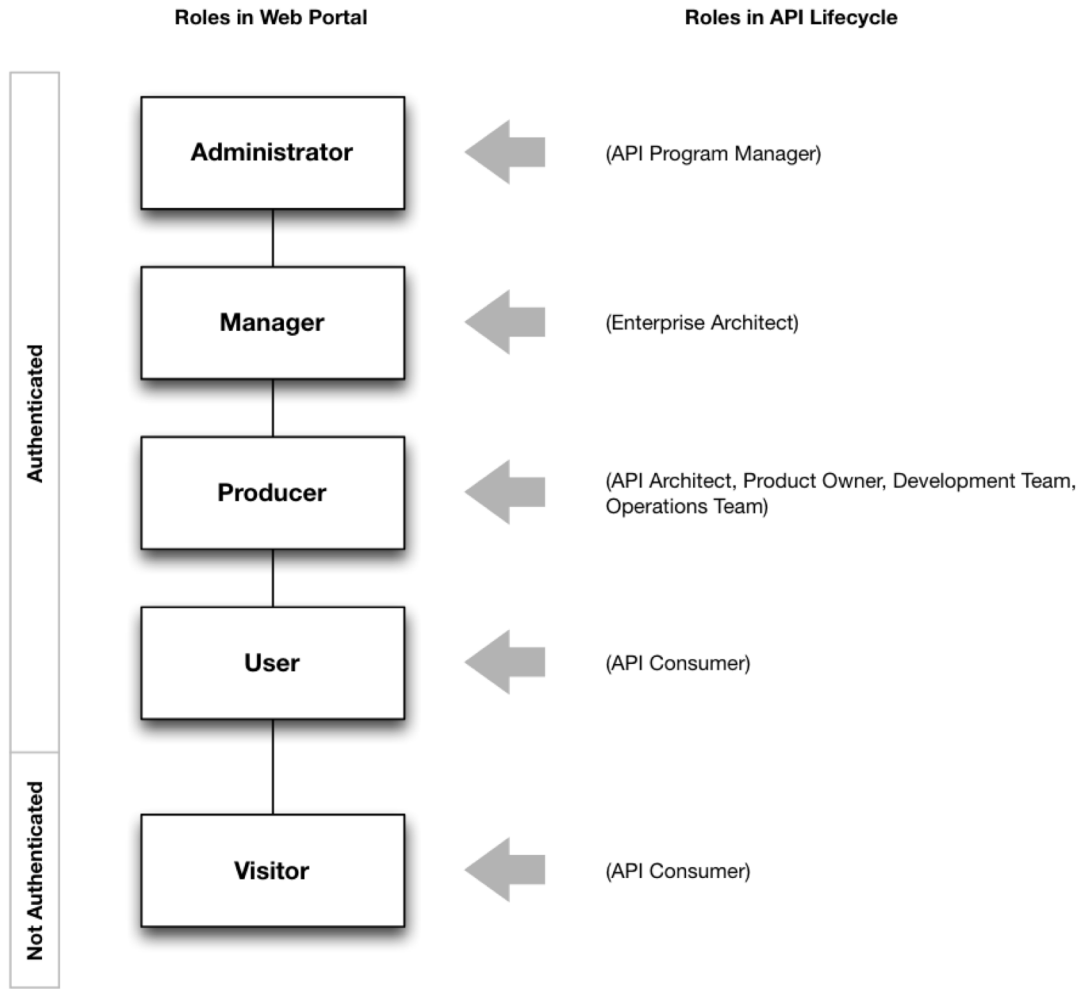


Table 4.1: The RBAC model for the web application prototype. (R: Role, P: Permission)

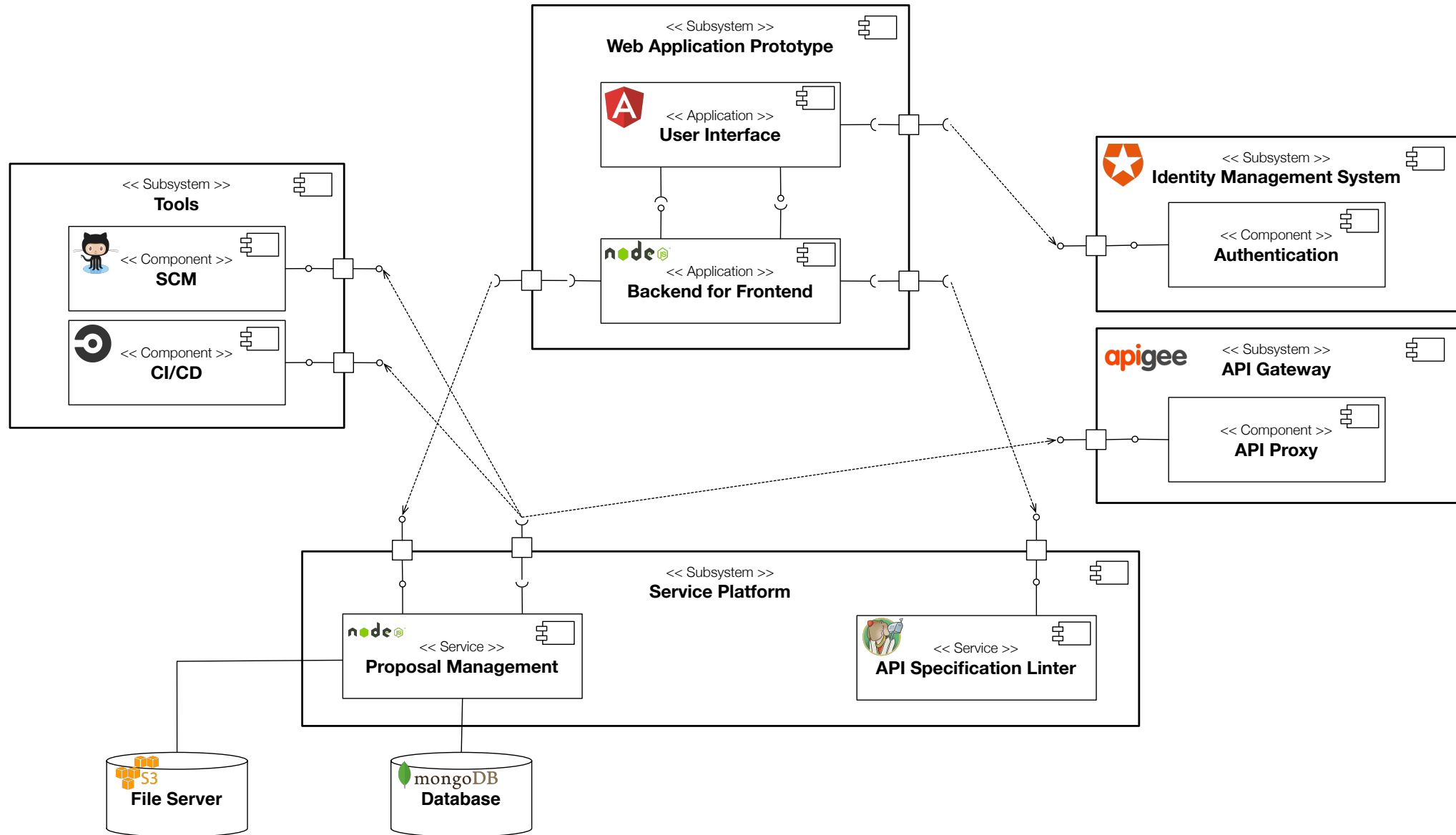
	P1: View API Proposal (Parts)	P2: View API Proposal (Full)	P3: Create API Proposal	P4: Update API Proposal	P5: Delete API Proposal	P6: Comment API Proposal	P7: Vote API Proposal	P8: Share API Proposal	P9: Get Approval for API Proposal	P10: Review Proposal	P11: View SLA	P12: Manipulate (Create, Update, Delete) SLA	P13: View Status of all APIs	P14: View Sandbox
R1: Visitor	x					x		x					x	x
R2: User	x	x*	x	x*		x	x	x			x*	x*	x	x
R3: Producer		x	x	x**		x	x	x			x	x**	x	x
R4: Manager		x	x	x**	x+	x	x	x	x+	x+	x	x**	x	x
R5: Administrator		x	x	x	x	x	x	x	x	x	x	x	x	x

*Role must be owner of the proposal

+Role must have the same domain as the proposal

**Role must be owner of the proposal or must have the same domain as the proposal

Prototype Architecture



Prototype

Data Model

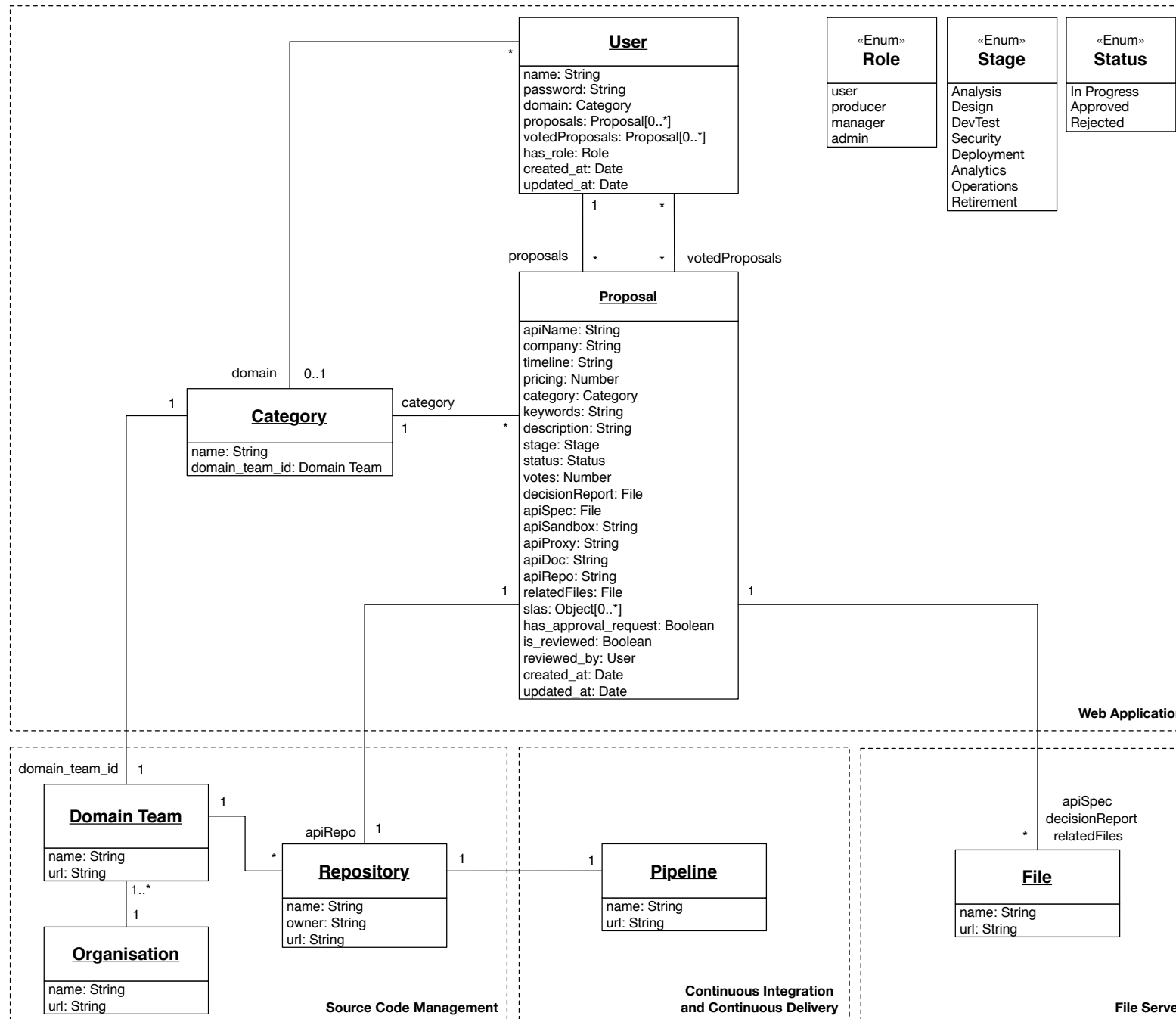


Table 6.1: The chosen interview partners for the evaluation of the prototype

ID	Role	Alias	Years Active	Core Area	Dev Experience
1	Research Associate	RA1	5 years	Knowledge Management, Knowledge Transfer, Software Architecture	7 years
2	Research Associate	RA2	3.5 years	Model based User Interfaces	20 years
3	Enterprise Architect	EA1	6 years	Big Data, Search Technologies, Distributed Systems	20 years
4	Enterprise Architect	EA2	8 years	System Integration, SOA	15 years
5	Software Architect	SW1	3 years	Infrastructure, DevOps	17 years
6	Enterprise Architect	EA3	8 months	Mobile Apps	10 years
7	Enterprise Architect	EA4	3.5 years	Application Integration, SOA	5 years