

## API-Gateway bringt Ordnung in Microservices-Wildwuchs



Frank Pientka, Dortmund

A photograph of the Golden Gate Bridge in San Francisco, California. The bridge is a suspension bridge with two large towers and numerous cables. The bridge is painted a distinctive orange-red color. The sky is a clear blue with some light, wispy clouds. The bridge spans across a body of water, and the foreground is partially obscured by a layer of white, fluffy clouds.

Warum sind Verbindungen  
wichtig?

# Agenda

- APIs regieren die Welt – Schnittstellenprobleme
- Microservices und Qualitätsmerkmale (NFRs)
- Microservices und Design Patterns
- Das API-Gateway Design Pattern
- API-Gateway Implementierungen
- Fazit



# The Great Baltimore Fire of 1904

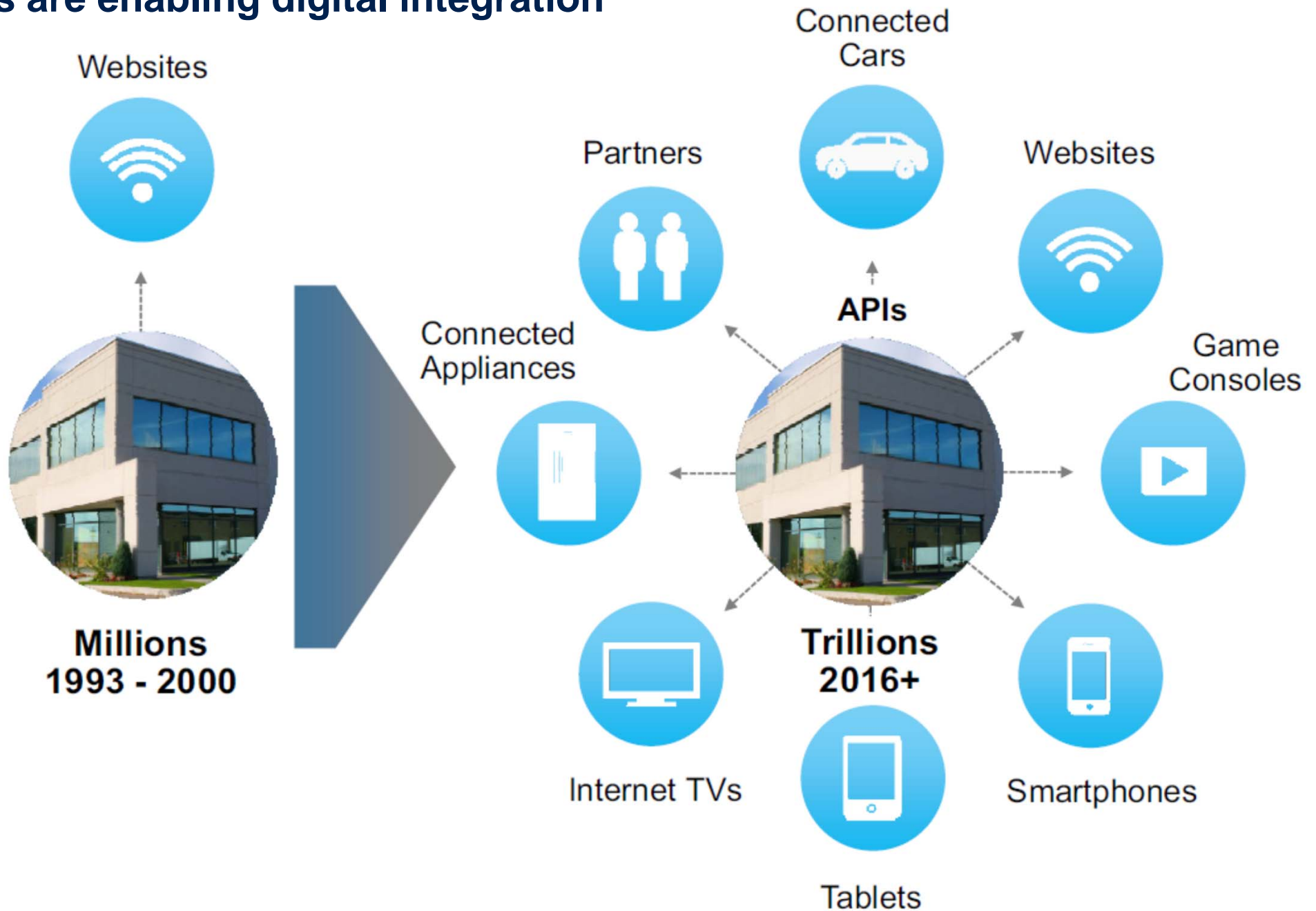




**Schnittstellenprobleme gestern, wie heute  
sind gefährlich und teuer ...**



# APIs are enabling digital integration





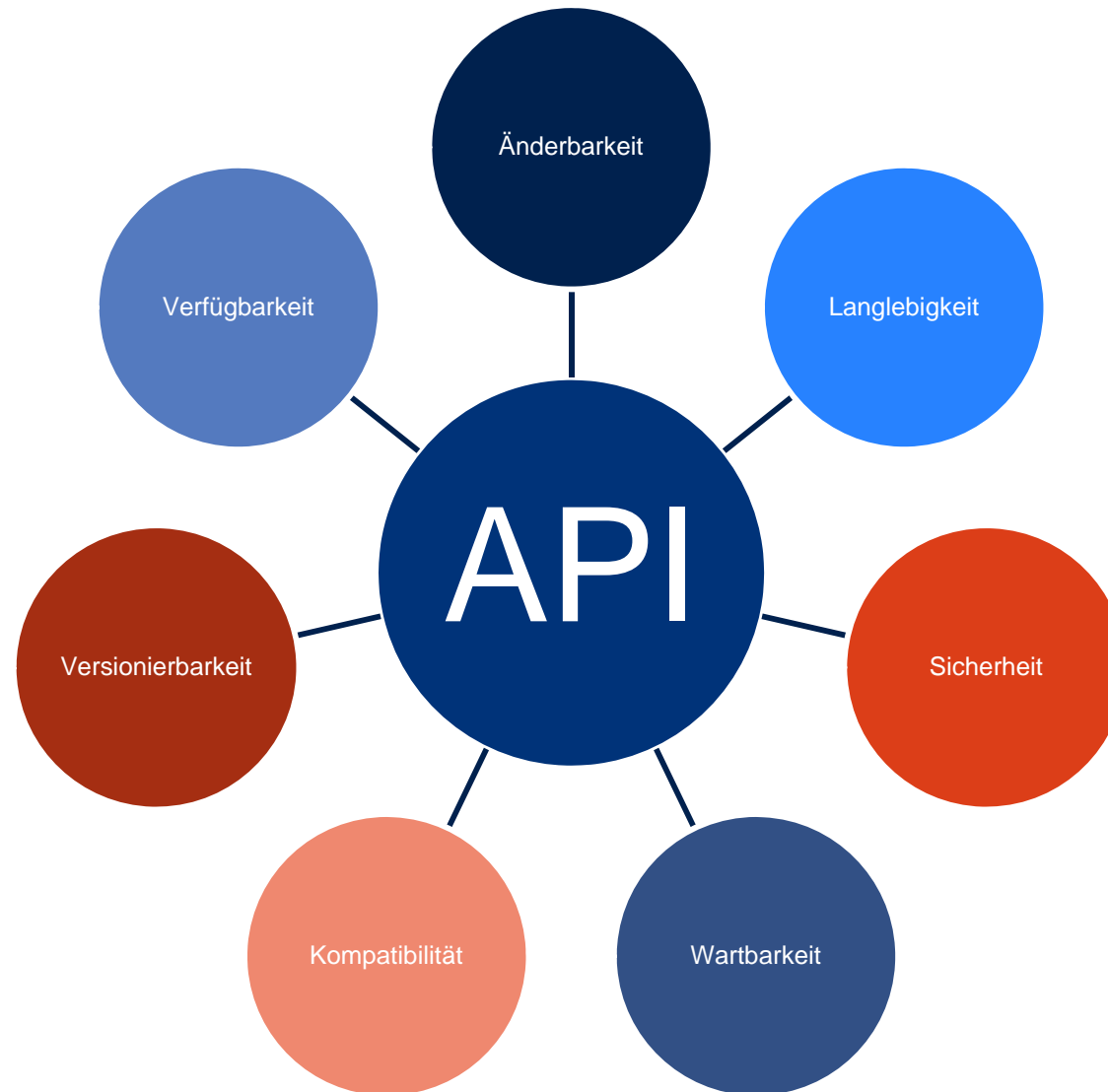
# Risikopotential APIs



- Schnittstelle
- Protokoll
- Implementierung
- Infrastruktur
- Betrieb
- Überwachung
- Änderung
  
- Hohe Kosten, Risiken
  
- Besonders Cloud, IoT

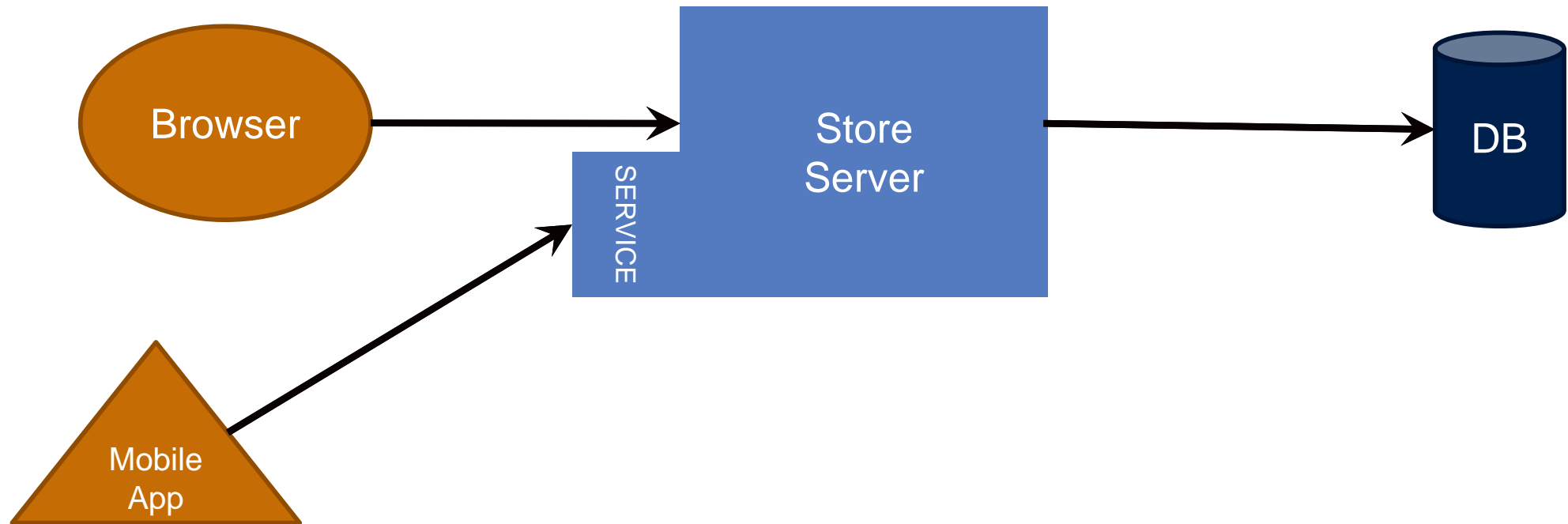
# Qualitätsanforderungen an eine API

Vertrag  
Design  
Governance  
Policy  
Authentifizierung  
Autorisierung  
Schutz  
Caching  
Throttling  
Quotas

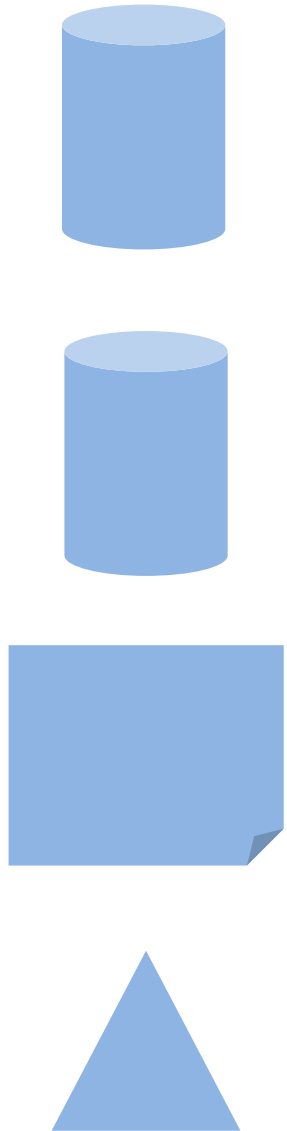




# Traditionelle Webarchitektur



# Microservices



Order Entry Service

Order History Service

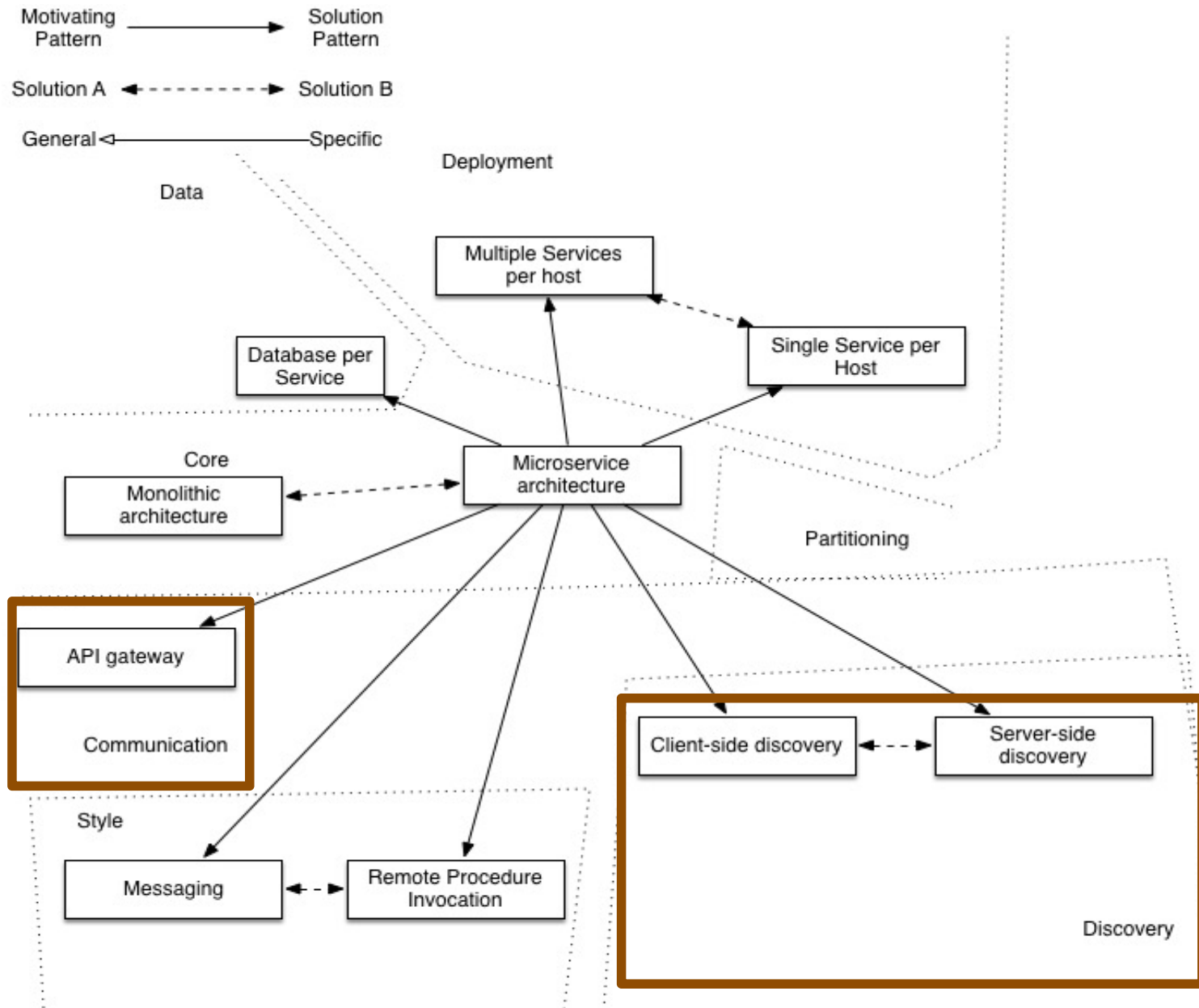
Product Search Service

Product Detail Service

Price Service



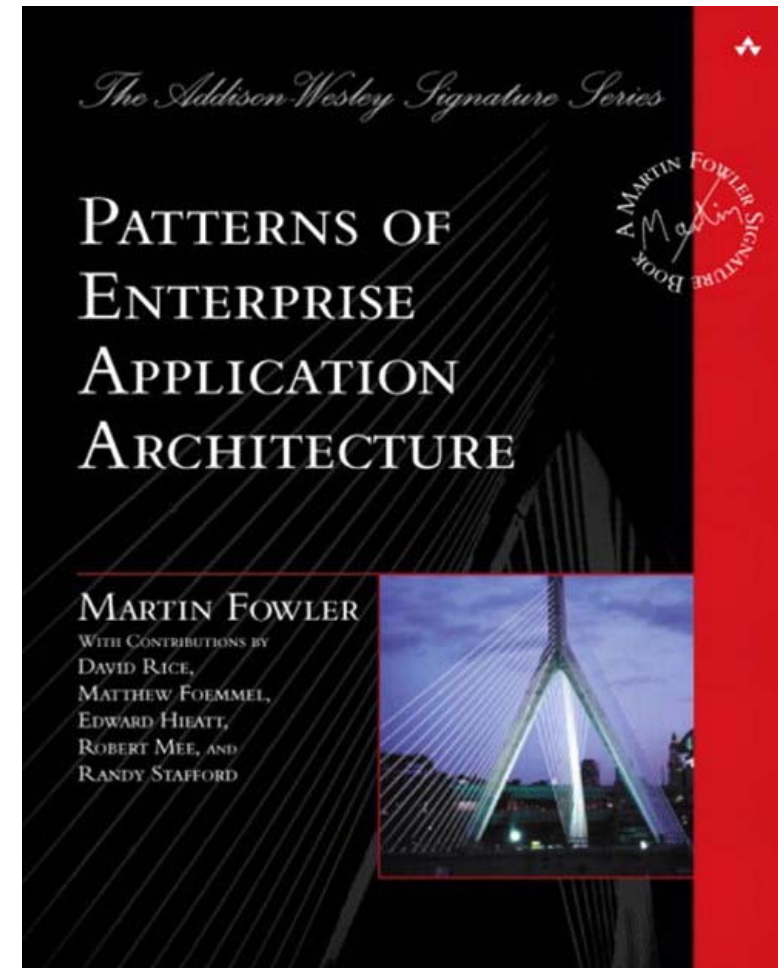
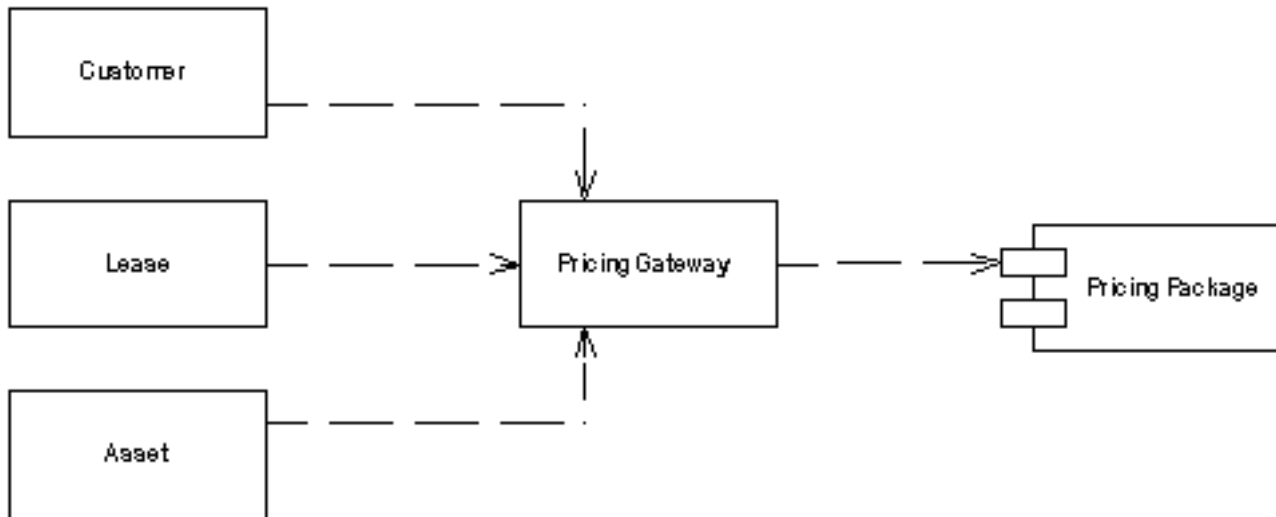
# Aus der Musterküche: Microservice Design Patterns (C. Richardson)



# Gateway Pattern of Enterprise Application Architecture (EAA) 2003

An object that encapsulates access to an external system or resource. (see EAA page 466)

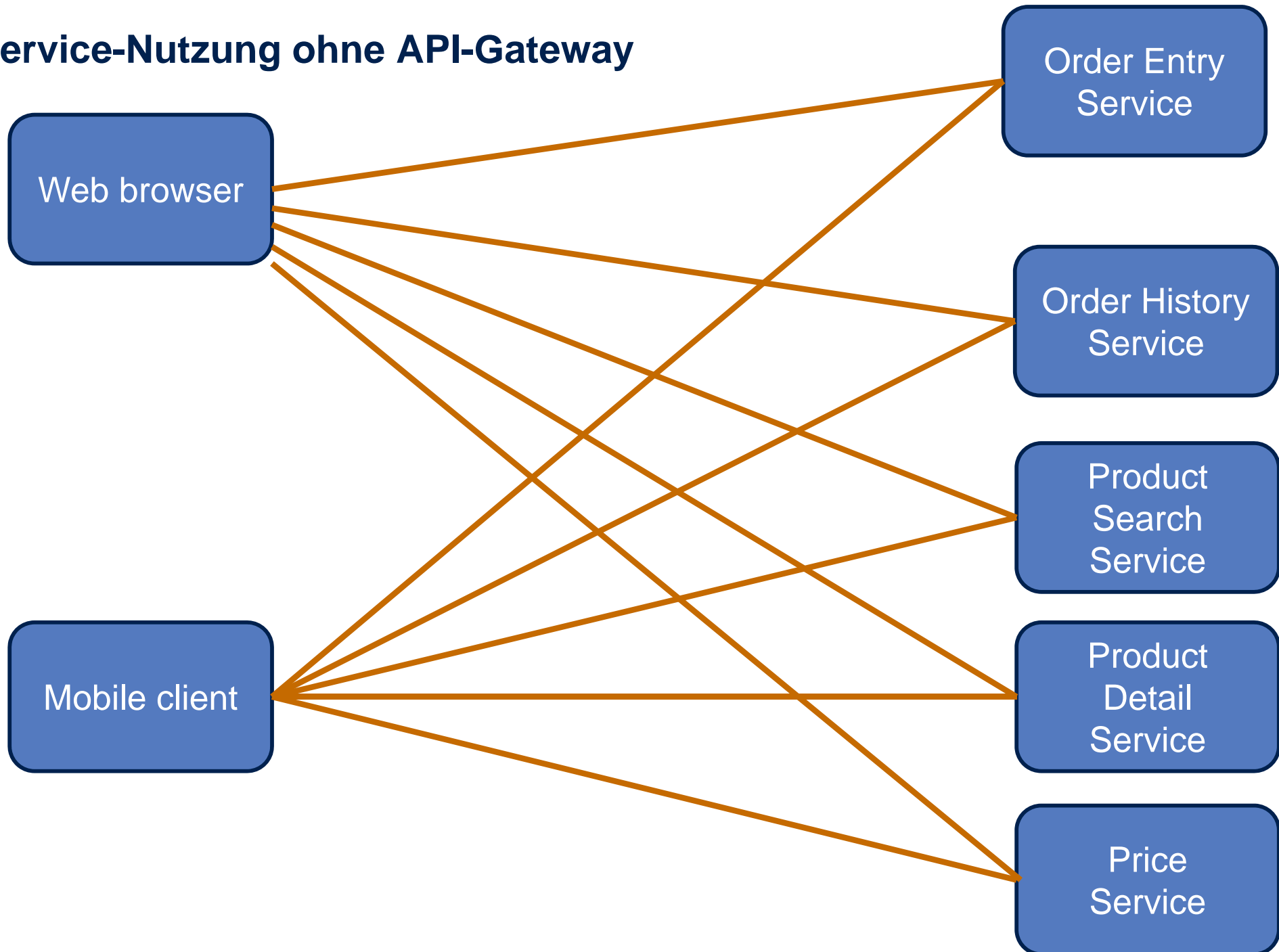
Similar Remote Façade, Filter, Router, Proxy



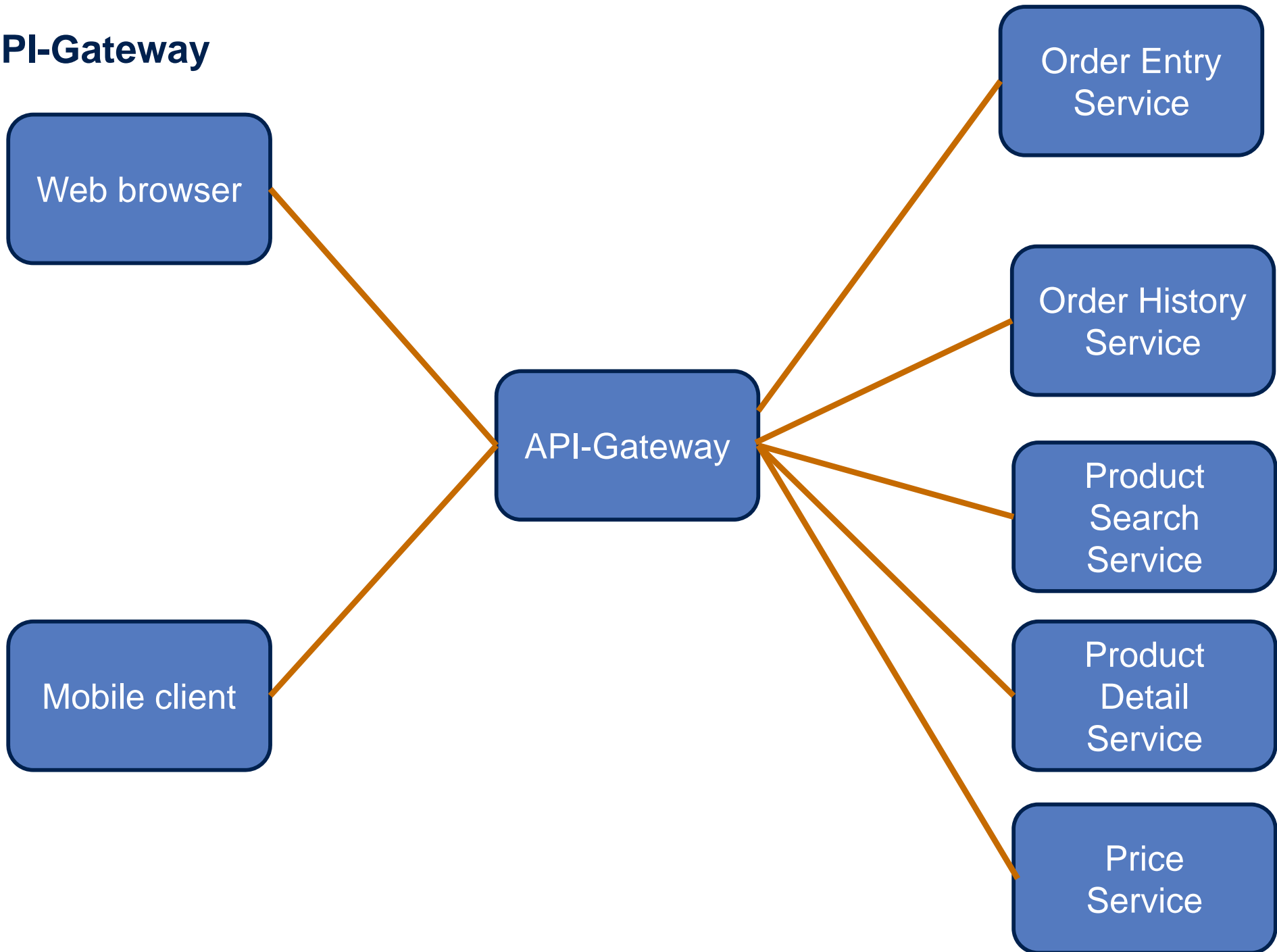
<http://martinfowler.com/eaCatalog/gateway.html>



## Service-Nutzung ohne API-Gateway

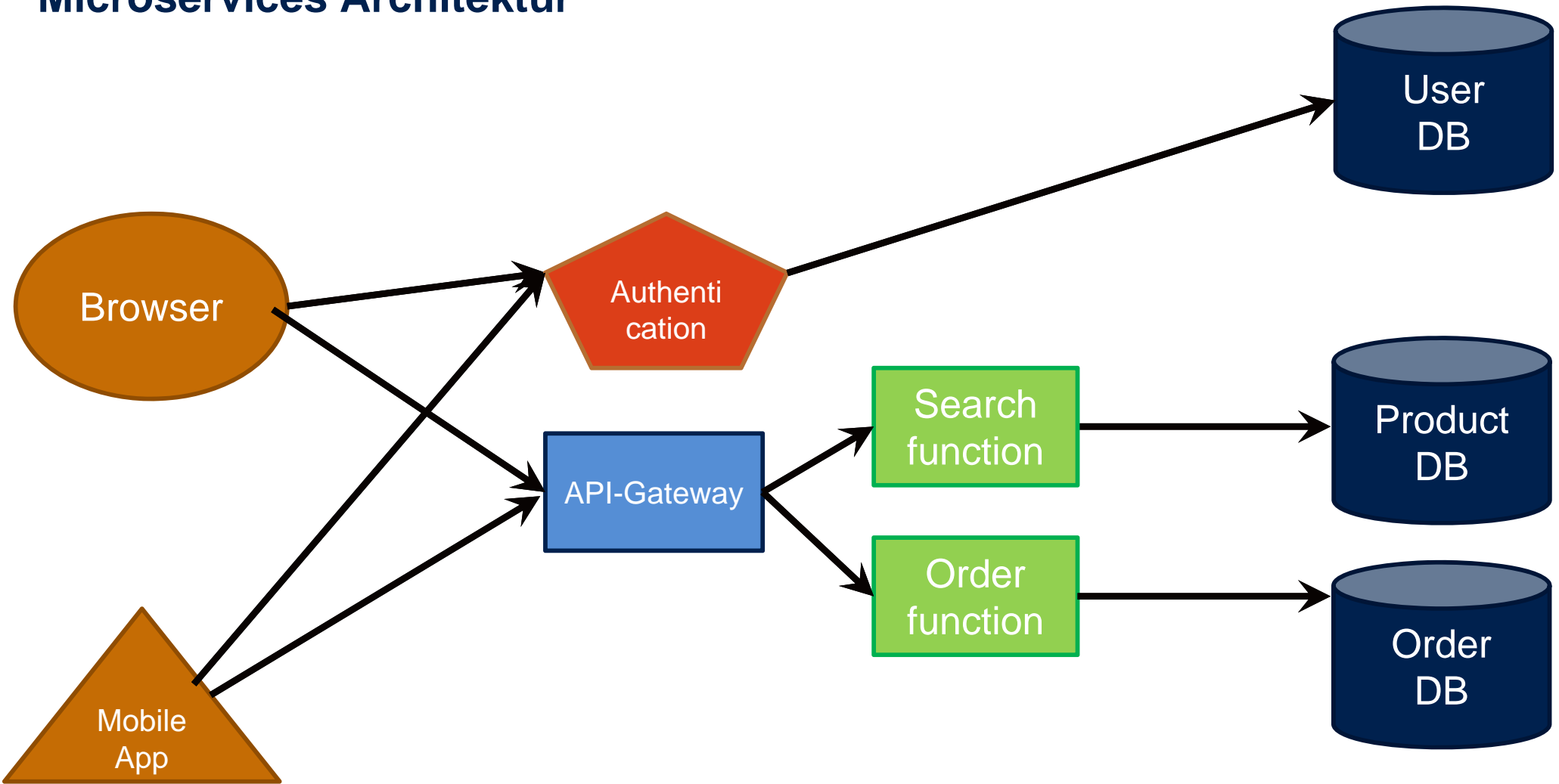


# API-Gateway





# Microservices Architektur



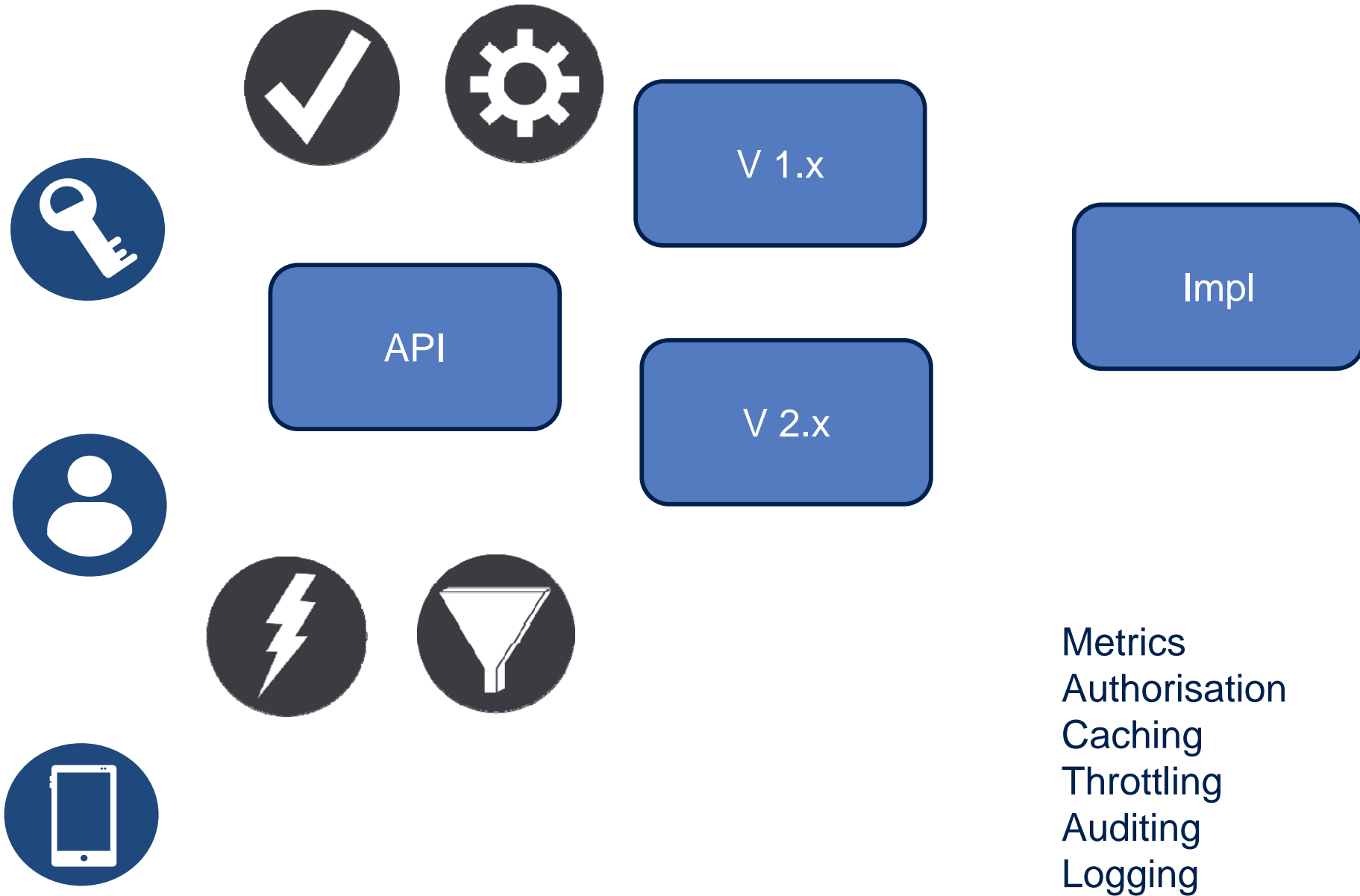
Serverless architectures

FaaS

BfF

BaaS

# API-Gateway – API-First Strategy



# Gartner Magic Quadrant for Application Services Governance

23.06.2016 Red Hat übernimmt API-Manager 3scale  
 09.09.2016 Google kauft API-Management-Provider Apigee



## API-Gateway-Lösungen: public vs. private



Amazon API Gateway



JBoss API Gateway



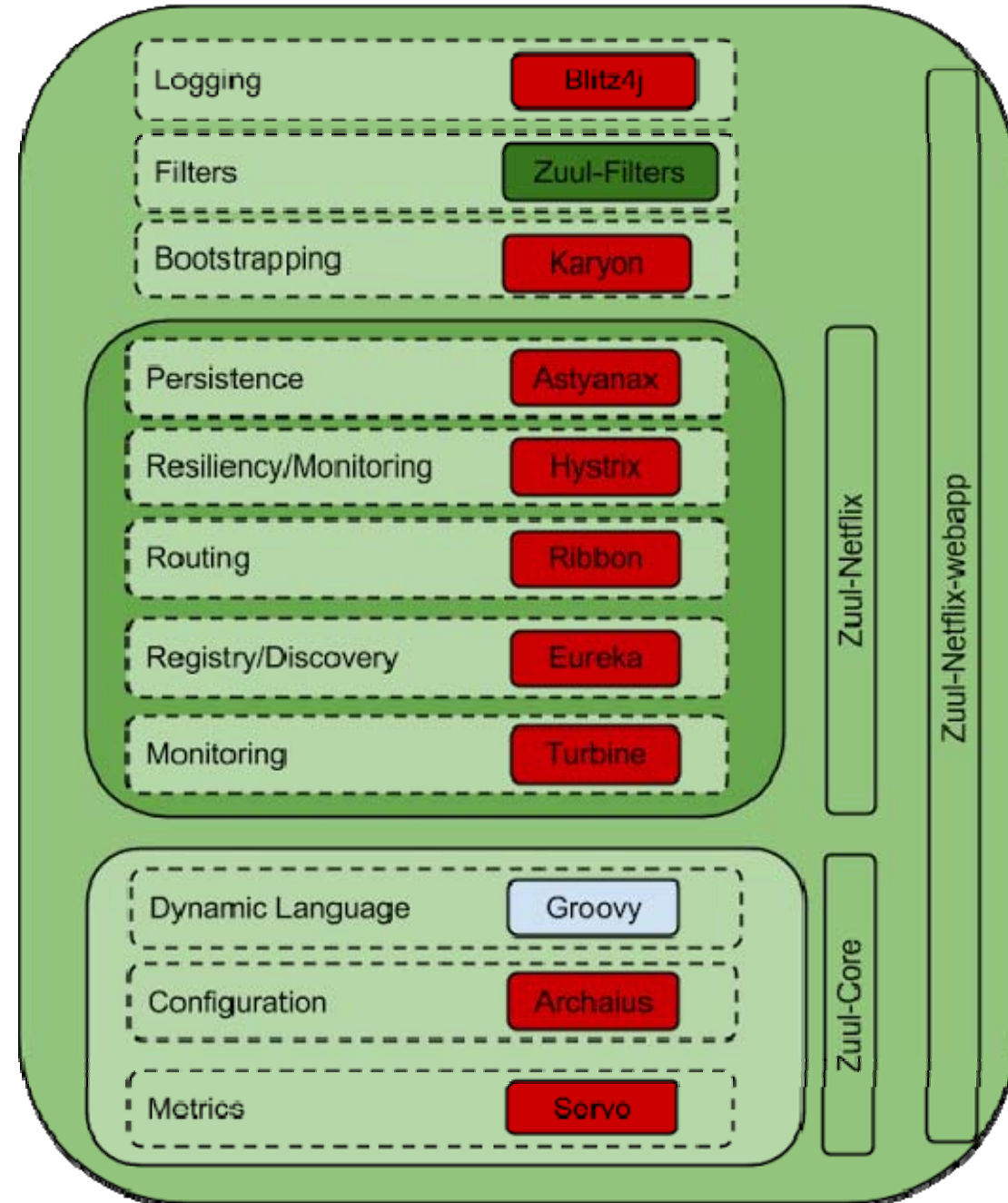
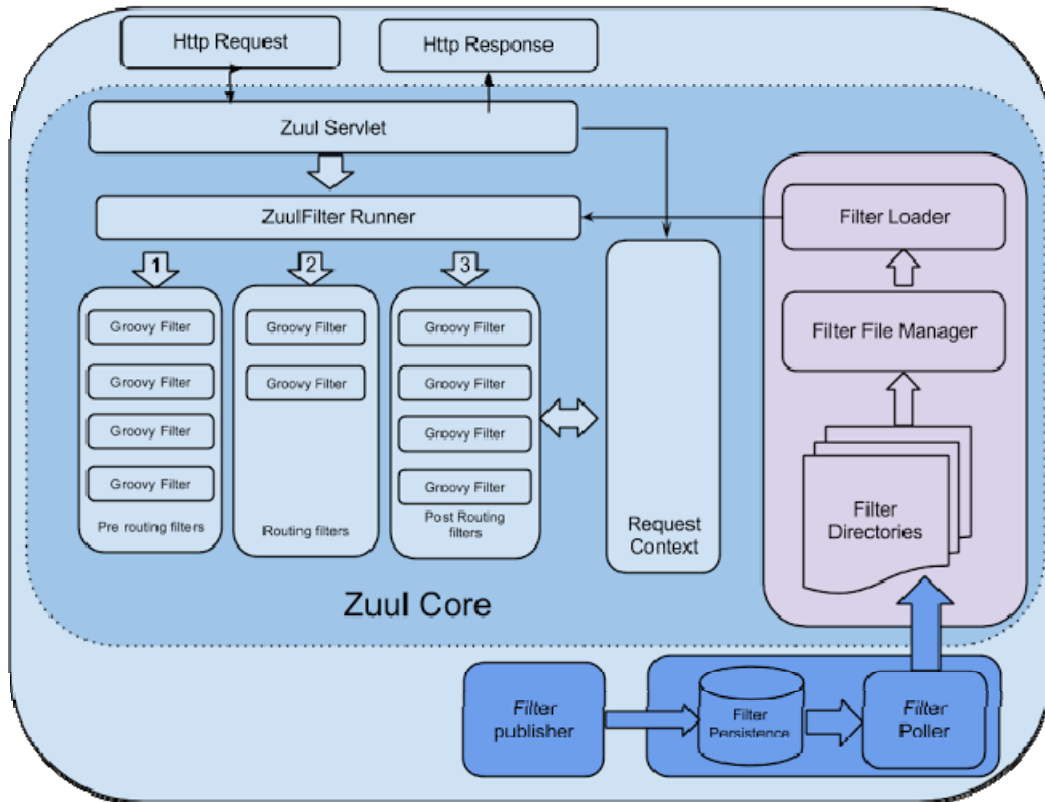
3scale's API Management platform



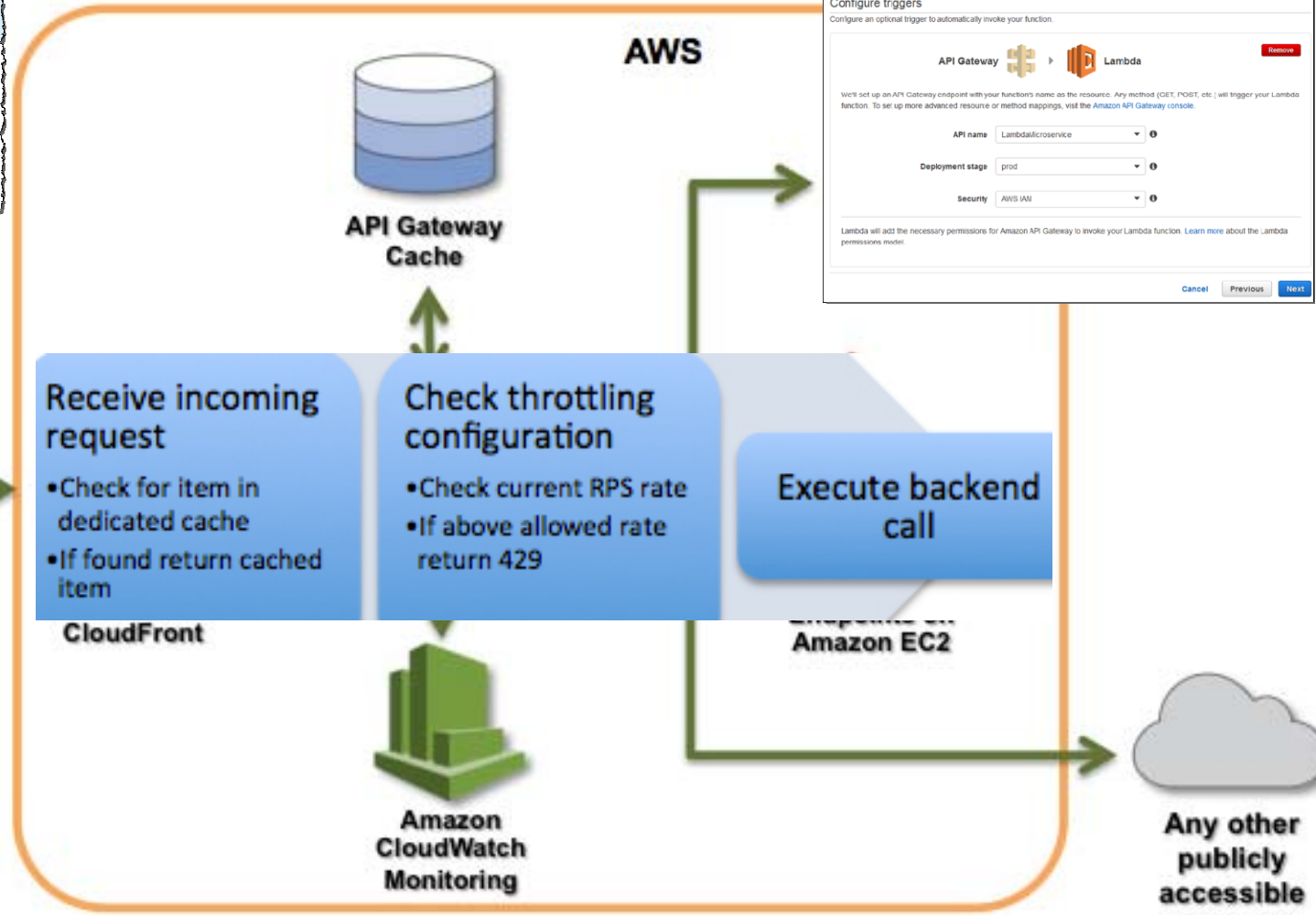
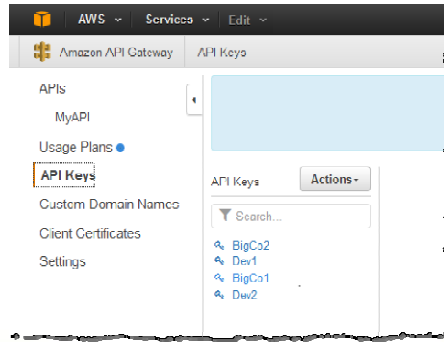
Netflix Zuul Proxy Filter



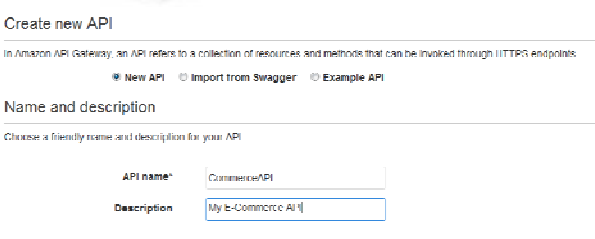
# Netflix Zuul: API Edge Service



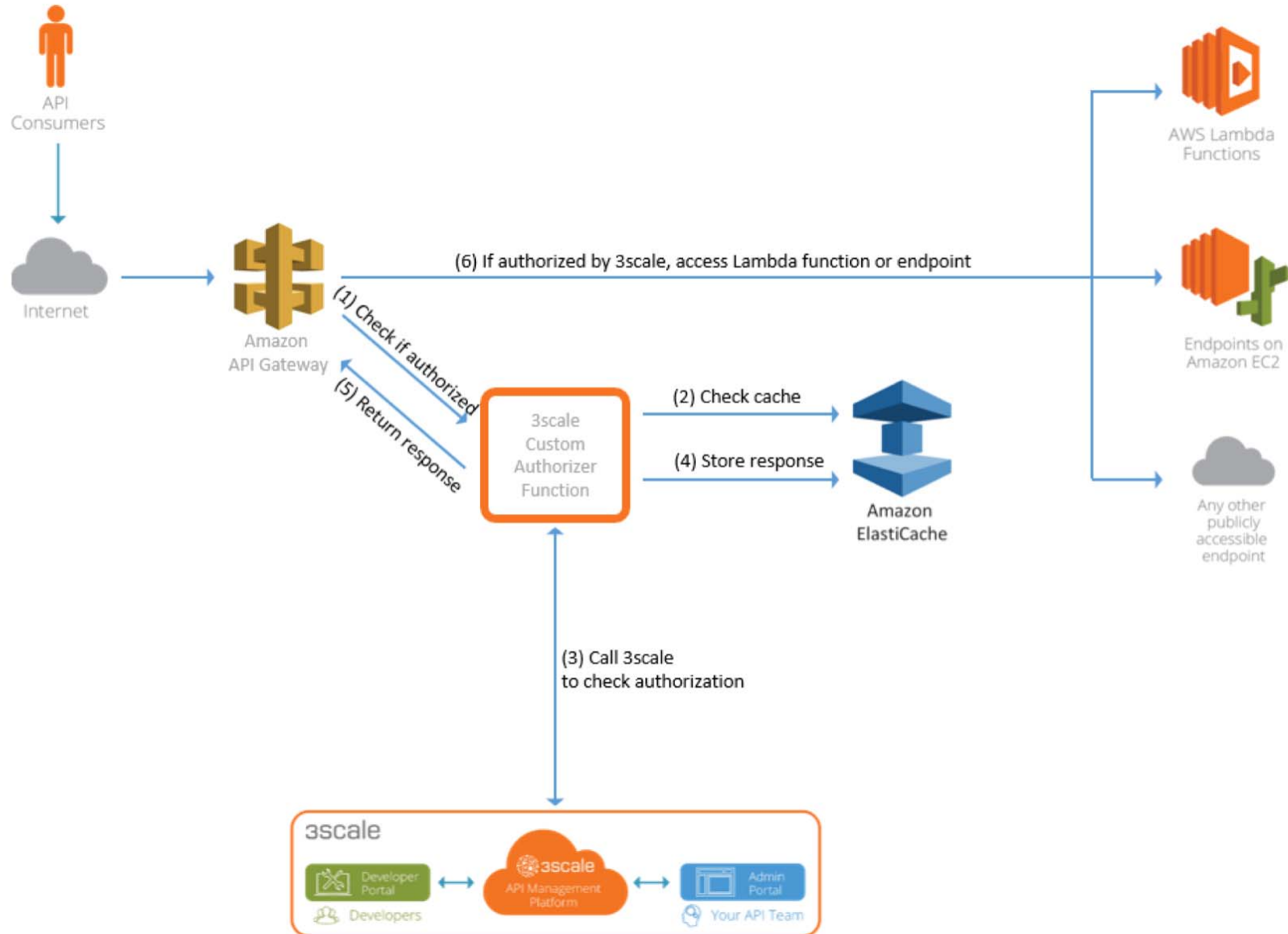
<https://aws.amazon.com/de/api-gateway/>



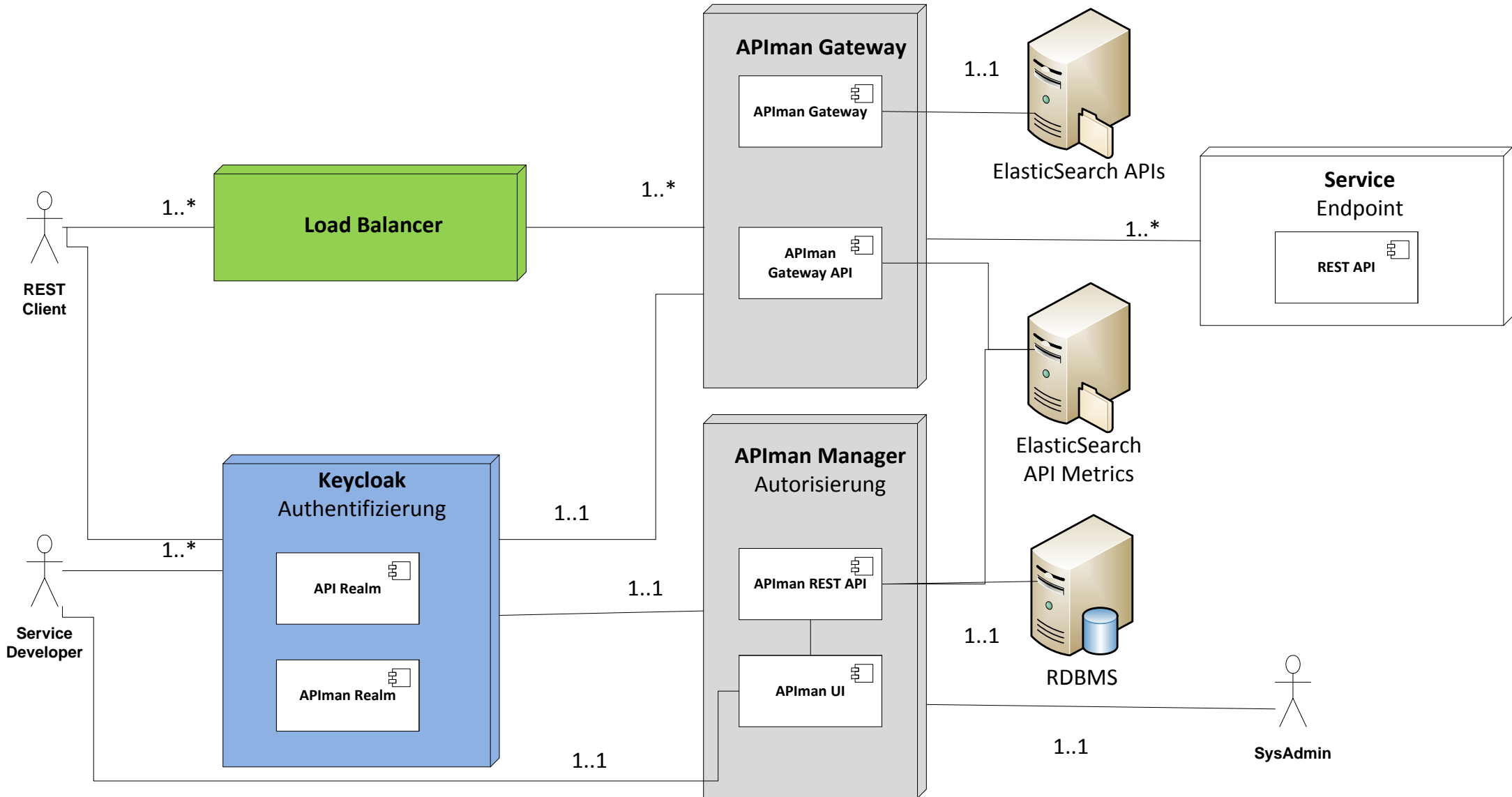
# An Amazon API Gateway Call Flow



# 3scale + Amazon API Gateway = Full Complement API Program Management



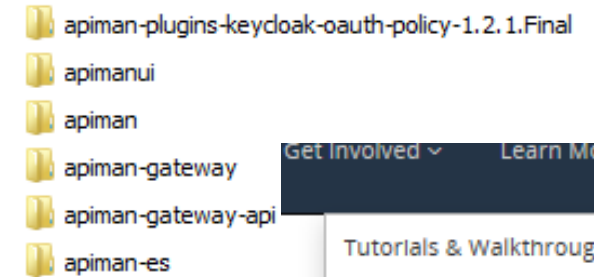
# APIman Architektur





# Getting Started APIman

```
<role rolename="apiuser"/>
<role rolename="apiadmin"/>
<role rolename="apipublisher"/>
<user username="admin" password="admin123!" roles="apiuser,apiadmin"/>
<user username="apimanager" password="apiman123!" roles="apipublisher"/>
</tomcat-users>
```



## Getting Started (WildFly)

One easy way to get started with **apiman** is to simply download WildFly 9 and the apiman overlay. Unpack them both into the same location and you're off and running. You'll obviously also need to have Java installed - we currently require at least version 8.

WildFly 10   WildFly 9   **Tomcat 8+**   EAP 7

### Download

- [Tomcat 8.0.33](#)
- [apiman 1.2.7.Final](#) overlay for Tomcat 8

### Install

1. Unpack the Apache Tomcat 8 zip
2. Unpack the apiman 1.2.7.Final Tomcat 8 overlay zip **inside the tomcat directory**

### Run

1. Start Tomcat 8 using the default configuration
2. Point your browser at the [apiman UI](#) and login with admin/admin123!

### Or simply try this...

```
mkdir ~/apiman-1.2.7.Final
cd ~/apiman-1.2.7.Final
curl http://apache.mirrors.tds.net/tomcat/tomcat-8/v8.0.33/bin/apache-tomcat-8.0.33.zip
curl http://downloads.jboss.org/apiman/1.2.7.Final/apiman-distro-tomcat8-1.2.7.Final-overlay.zip
unzip apache-tomcat-8.0.33.zip
unzip -o apiman-distro-tomcat8-1.2.7.Final-overlay.zip -d apache-tomcat-8.0.33
cd apache-tomcat-8.0.33
chmod 755 bin/catalina.sh
./bin/catalina.sh run
```

localhost:8080/apimanui/ → 1.2.6 http://localhost:8080/apiman conf/apiman.properties, tomcat-users.xml

# API Management

Welcome to apiman, open source API management. Use this software to provide various APIs (APIs) to your consumers (Client Apps) through a secure, scalable, and governed Gateway layer. API Management allows API Developers to centralize control and analysis of their APIs. It also provides a central location for Client App Developers to find and consume available APIs.

## Organizations

All APIs and Client Apps must be managed within the context of an Organization. You can be a member of multiple Organizations at the same time, with different roles in each: you can be a Client App Developer in one organization and an API Developer in another.

- [Create a New Organization](#)
- [Go to My Organizations](#)
- [Browse/Find an Organization](#)

## APIs

Create, find, or manage APIs. An API is also known as a "Service" - anything that can be invoked remotely by some sort of client application. This platform simply provides a way to turn unmanaged (raw) back-end APIs into "managed" APIs by attaching policies to them.

- [Manage My APIs](#)
- [Create a New API](#)
- [Browse available/importable APIs](#)

## Client Apps

Create and manage your Client Apps. A Client App is the thing that consumes APIs. These consumers must be defined in this platform so that contracts can be created between them and the APIs they wish to consume.

- [Manage My Client Apps](#)
- [Create a New Client App](#)
- [Find/Consume an API](#)

## System Administration

Hey it looks like you're an administrator! Here are some things only you can do. These are system-wide settings you're thinking about modifying, so please proceed with caution.

- [Manage Roles/Permissions](#)
- [Manage Policy Definitions](#)
- [Manage Gateways](#)
- [Manage Plugins](#)
- [Export/Import Data](#)

# API Catalog









[Home](#) » [API Catalog](#)

## API Catalog

List of APIs available to be imported. These APIs were retrieved from the configured API Catalog.

Filter by API name or type...

 Amazon S3 API <a href="#">Import</a>   <a href="#">Definition</a>	 Facebook API <a href="#">Import</a>   <a href="#">Definition</a>	 Flickr API <a href="#">Import</a>   <a href="#">Definition</a>	 Google Maps API <a href="#">Import</a>   <a href="#">Definition</a>
 Twitter API <a href="#">Import</a>   <a href="#">Definition</a>	 YouTube API <a href="#">Import</a>   <a href="#">Definition</a>		



### Twitter API

The Twitter micro-blogging service includes two RESTful APIs.

Created on 2016-02-02  
Created by admin

Status: **READY**

Version: 1.1 ▾ **New Version**

- 1.1
- 1.0

Create a new version of this API (New Version)

**Publish**

Overview

Implementation

Definition

Plans

Policies

Activity

## API Details

This is the API details page. Use this page to modify the meta-data, plans, and policies for the API. There is no need to follow the tabs in order, but note that you will need to fill out a minimum amount of data before the API can be published to the Gateway. In particular, the Implementation and Plans tabs are important (more information below).

### Implementation

The 'Implementation' tab is where you go to configure the details of the real API being managed. This is where you tell apiman where the API is, so that we can actually invoke it at runtime.

### Definition

The 'Definition' tab allows you to configure an optional API Definition document for your API. A API Definition is a detailed description of the capabilities of your API (operation names, http verbs, data models). For example, you might add a Swagger spec document as your API Definition.

### Plans

The 'Plans' tab is used to configure which Plans (already defined in the Organization) are available when a Client App wishes to create an API Contract with this API. Optionally, an API can also be marked as "Public", which means that no API Contract is required (the API can be invoked without sending an API Key).

### Policies

The 'Policies' tab is where you go if you wish to configure any policies that should be enforced for \*all\* requests to the API, regardless of which user, Client App, or Plan is being used. In other words, API-level policies.

### Contracts

The 'Contracts' tab will show a table of all Client Apps that have created API Contracts to this API. In other words, it shows all of the registered consumers of the API. Note that if the API is marked as "Public", this tab will likely be empty.

### Endpoint



# Vertrag erstellen



## New Contract

Creating a Contract allows you to connect a Client App to an API via a particular Plan offered by the API. You would want to do this so that your Client App can invoke the API successfully. Note that this is not necessary if the API is public.

### From Client App

The Client App that will be used as the source of the new API Contract. Choose one of your available Client Apps below, and then choose a Client App version.

MATERNA / mobileapp ▾

1.0 ▾

### Using Plan

Please choose a valid API below before selecting the Plan you wish to use for this Contract (either you have not selected an API or the API has no available plans).

### To API

Use this section to choose what API the Client App will be consuming (aka the "target" of this API Contract).

MATERNA / mobileapi ⇄ 1.0

(click to change)

Create Contract

Cancel

# Apiman Plugins



Roles

Policy Definitions

Gateways

Plugins

Export/Import

## Plugins

Manage the plugins known to this installation of apiman. Plugins allow additional functionality to be included in the system after it has been installed and configured.

Installed Plugins

Available Plugins

Filter by plugin name...



Add Custom Plugin

Name	Description	Actions
CORS Policy Plugin	This plugin implements CORS (Cross-origin resource sharing): A method of controlling access to resources outside of an originating domain.	<a href="#">Install</a>
HTTP Security Policy Plugin	Provides a policy which allows security-related HTTP headers to be set, which can help mitigate a range of common security vulnerabilities.	<a href="#">Install</a>
JSONP Policy Plugin	A plugin that contributes a policy that turns a standard RESTful endpoint into a JSONP compatible endpoint.	<a href="#">Install</a>
Keycloak OAuth Policy Plugin	This plugin offers an OAuth2 policy which leverages the Keycloak authentication platform as the identity and access provider.	<a href="#">Install</a>
Log Headers Policy	A policy that logs the headers to std out. Useful to analyse inbound HTTP traffic to the gateway when added as the first policy in the chain or to analyse outbound HTTP traffic from the gateway when added as the last policy in the chain.	<a href="#">Install</a>
Simple Header Policy Plugin	Offers a simple policy that allows request headers to be added or stripped from the HTTP request (outgoing) or HTTP response (incoming).	<a href="#">Install</a>
XML<->JSON Transformation Policy Plugin	This plugin provides a very simple policy which can transform the request and/or response payload between XML and JSON.	<a href="#">Install</a>

# Apiman Policy




- Roles
- Policy Definitions
- Gateways
- Plugins
- Export/Import

## Policy Definitions


Configure the available policy definitions. These will be the policies made available to users when configuring clients, APIs, and plans.

Filter by policy name...


[Import Policy](#)

-  **Authorization Policy**  
 Enables fine grained authorization to API resources based on authenticated user roles.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.AuthorizationPolicy


---

-  **BASIC Authentication Policy**  
 Enables HTTP BASIC Authentication on an API. Some configuration required.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.BasicAuthenticationPolicy


---

-  **Caching Policy**  
 Allows caching of API responses in the Gateway to reduce overall traffic to the back-end API.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.CachingPolicy


---

-  **IP Blacklist Policy**  
 Requests that originate from a specified set of valid IP addresses will be denied access.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.IPBlacklistPolicy


---

-  **IP Whitelist Policy**  
 Only requests that originate from a specified set of valid IP addresses will be allowed through.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.IPWhitelistPolicy


---

-  **Ignored Resources Policy**  
 Requests satisfying the provided regular expression will be ignored.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.IgnoredResourcesPolicy


---

-  **Quota Policy**  
 Provides a way to limit the total number of requests that can be sent to an API.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.QuotaPolicy

---

-  **Rate Limiting Policy**  
 Enforces rate configurable request rate limits on an API. This ensures that consumers can't overload an API with too many requests.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.RateLimitingPolicy

---

-  **Transfer Quota Policy**  
 Provides a way to limit the total number of bytes that can be transferred from (or to) an API.  
**Implementation:**  
 class:io.apiman.gateway.engine.policies.TransferQuotaPolicy

# APIman Systemstatus

<http://www.apiman.io/latest/api-manager-restdocs.html>



miredot

Resources Search

expand / collapse all

- /
- actions POST
- downloads/{downloadId} GET
- currentuser
- gateways
  - / GET PUT POST
  - {gatewayId} GET PUT DELETE
- organizations
- permissions
- plugins
- policyDefs
- roles
- search
- system
  - export GET
  - import POST
  - status GET
- users

API Manager REST Services (1.2.2.Final)

## GET SYSTEM STATUS

This endpoint simply returns the status of the apiman system. This is a useful endpoint to use when testing a client's connection to the apiman API Manager REST services.

**GET** <http://localhost:8080/apiman/system/status/>

### Returns

System status information.

Content-Type: `application/json`

```
{
  name: string
  id: string
  version: string
  description: string
  up: boolean
  builtOn: string
  moreInfo: string
}
```

Hide descriptions

### Status codes

200 OK

On success:

API Manager REST Services (1.2.6.Final)

## EXECUTE AN ENTITY ACTION

Call this endpoint in order to execute actions for apiman entities such as Plans, APIs, or Clients. The type of the action must be included in the request payload.

**POST** <http://localhost:8080/apiman/actions/>

### Body

The details about what action to execute.

Accept: `application/json`

```
{
  type: publishAPI | retireAPI | registerclient | unregisterclient | lockPlan
  entityId: string
  entityVersion: string
  organizationId: string
}
```

Hide descriptions

### Status codes

204 No Content If the action completes successfully.

## DOWNLOAD FILE

This endpoint is used to download a file that was previously generated by some other REST API call. For example, when exporting data via the /system/export endpoint, a temporary download link may be created. This represents that temporary download link. In this example, the download will result in the exported data.

**GET** <http://localhost:8080/apiman/downloads/downloadId/>

### Path parameters

downloadId comment missing  
string



# apiman.plugins.repositories=http://mvnrepository.com/artifact/io.apiman.plugins

[Home](#) > [System Administration](#)

[Roles](#)

[Policy Definitions](#)

[Gateways](#)

**[Plugins](#)**

[Export/Import](#)

## Plugins

Manage the plugins known to this installation of apiman. Plugins allow additional functionality to be included in the system after it has been installed and configured.

Installed Plugins

Available Plugins

Filter by plugin name...



Add Custom Plugin

Name	Description	Actions
Circuit Breaker Policy	This plugin has policies for performing circuit breaker functionality.	<a href="#">Install</a>
CORS Policy Plugin	This plugin implements CORS (Cross-origin resource sharing): A method of controlling access to resources outside of an originating domain.	<a href="#">Install</a>
HTTP Security Policy Plugin	Provides a policy which allows security-related HTTP headers to be set, which can help mitigate a range of common security vulnerabilities.	<a href="#">Install</a>
JSONP Policy Plugin	A plugin that contributes a policy that turns a standard RESTful endpoint into a JSONP compatible endpoint.	<a href="#">Install</a>
Keycloak OAuth Policy Plugin	This plugin offers an OAuth2 policy which leverages the Keycloak authentication platform as the identity and access provider.	<b>INSTALLED</b>
Log Headers Policy	A policy that logs the headers to std out. Useful to analyse inbound HTTP traffic to the gateway when added as the first policy in the chain or to analyse outbound HTTP traffic from the gateway when added as the last policy in the chain.	<b>INSTALLED</b>
Simple Header Policy Plugin	Offers a simple policy that allows request headers to be added or stripped from the HTTP request (outgoing) or HTTP response (incoming).	<a href="#">Install</a>
XML<->JSON Transformation Policy Plugin	This plugin provides a very simple policy which can transform the request and/or response payload between XML and JSON.	<a href="#">Install</a>



# History, Policies & Metrics

## API Activity

The list below is all of the activity (configuration changes made by apiman users) associated with this API.

admin added a policy to MATERNA / manager version 1.0.

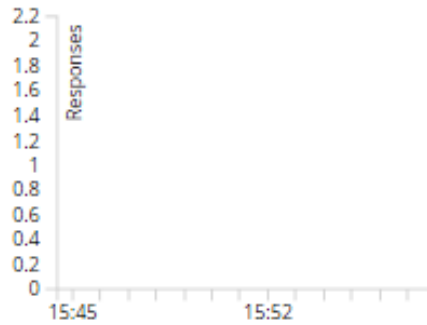
Policy added: IPWhitelistPolicy

2016-09-19 @ 4:47:25 PM

## API Metrics



Now that this API is published to the gateway, it *may* be consumed by API clients. Use this tab to view basic metrics/analytics for the API.

Show response type ▾ metrics data for the last hour ▾ Refresh

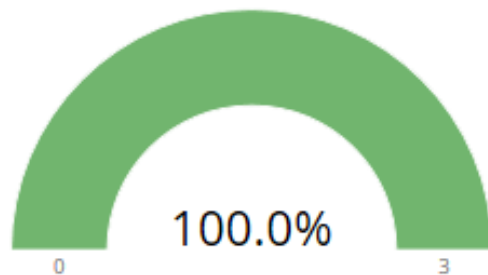


### API Policies

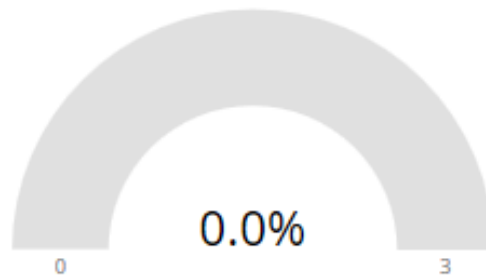
Here is a list of all Policies defined for this API. These Policies will be applied to all invocations of this API by any Client App, regardless of the Plan used in its Contract.

Policy Name	Description	Created By	Created On	Actions
 <b>Caching Policy</b>	Policy created by admin on 2016-09-19 API responses will be cached for 5 seconds.	admin	2016-09-19	<span>Add Policy</span> <span>Remove</span>
 <b>IP Whitelist Policy</b>	Policy created by admin on 2016-09-19 Only requests that originate from the set of 1 configured IP address(es) will be allowed to invoke the managed API.	admin	2016-09-19	<span>Remove</span>

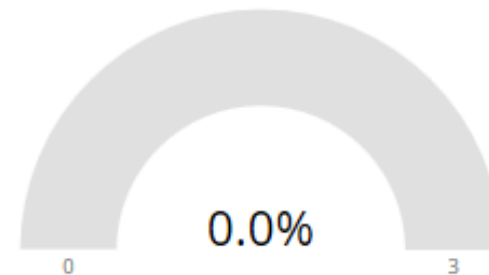
### Successful Responses



### Failed Responses



### Error Responses



# DEMO: Verwalten der Tomcat Manager REST-Schnittstelle mit APIman

Self managed versioned system ☺

Apache Tomcat/8.0.31



If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:

[Security Considerations HOW-TO](#)

[Manager Application HOW-TO](#)

[Clustering/Session Replication HOW-TO](#)

Server Status

Manager App

Host Manager

<http://localhost:8080/manager/text/list>

/apimanui:running:0:apimanui

/apiman-plugins-keycloak-oauth-policy-1.2.1.Final:running:0:apiman-plugins-keycloak-oauth-policy-1.2.1.Final

/apiman-gateway-api:running:0:apiman-gateway-api

/apiman:running:0:apiman

/apiman-es:running:0:apiman-es

/apiman-gateway:running:0:apiman-gateway

/manager:running:1:manager

# DEMO: Tomcat manager App versionieren

<http://localhost:8080/manager/text/list>

<https://localhost:8443/apiman-gateway/MATERNA/manager/1.0>

Overview

Implementation

Definition

Plans

Policies

Contracts

Endpoint

## API Implementation

Please provide us with details about the back-end API implementation so that the API Gateway can successfully proxy API requests. Please include any security you wish to enable between the API Gateway and the back-end API.

API Endpoint:

API Type:

API Content Type:

API Security:

[Home](#) > [MATERNA](#) > [manager](#)



Organizations Client Apps **APIs**

MATERNA / [Google Maps API](#)

The Google Maps API allow for the embedding of Google Maps onto web pages of outside developer.

MATERNA / [Twitter API](#)

The Twitter micro-blogging service includes two RESTful APIs.

MATERNA / [api](#)

MATERNA / [manager](#)

MATERNA / [mobileapi](#)

<no description>

🕒 Created on 2016-09-19

👤 Created by admin

Status: **PUBLISHED**

Version: 1.0 ▾

[New Version](#)

[Link my Client App to this API \(New Contract\)](#)  
[Create a new version of this API \(New Version\)](#)

[Re-Publish](#)

[Retire](#)

Overview

Implementation

Definition

## Managed Endpoint Information

To successfully invoke this managed API, a client must send the request to the appropriate API Gateway endpoint. When invoking the API through an API Contract, a valid API Key must be included in each request. If the API is public, it can be invoked directly (without an API Key) through the endpoint below.

### Managed Endpoint

## API-Gateway – horcht was kommt von draußen rein





**„Ein guter Zaun schafft gute Nachbarn“ Robert Frost**

*„Ein guter Vertrag schafft gute Nachbarn“*



# Ausblick



- Aktuell API-man mehr zum Evaluieren - 3scale-OS-Version abwarten
- Qualitätsanforderungen steigen
- Stärkere Entkopplung (Aufruf, Ressource)
- Unterschiedliche API-Granularitäten für unterschiedliche Clients
- NFRs zentral und einheitlich NICHT lokal in jedem Microservice
- API-(Lifecycle-)Management wird immer wichtiger (IoT, Cloud, mobile, Web, BfF)
- API-Gateway erzeugt neue Komplexität, verbessert jedoch Service-Management und -Nutzung

## Weitere Infos

- API Gateway design pattern, Chris Richardson <http://microservices.io/patterns/apigateway.html>
- APIman <http://www.apiman.io>
- Keycloak Reference Guide <http://www.keycloak.org>
- APIman Roadmap <http://www.apiman.io/latest/roadmap.html>
- Frank Pientka: Horcht, was kommt von draußen rein - API-Management mit apiman, JavaSPEKTRUM 07/16

