



Service Engineering

Technische Aspekte für (Web-) Services – Security, GraphQL

Die Inhalte der Vorlesung wurden primär auf Basis der angegebenen Literatur erstellt.
Darüber hinaus finden sich vielfältige Beispiele aus dem industriellen Umfeld.



Agenda



- Secure Web-APIs
- GraphQL
- Alternative Kurzübungen



Secure Web-APIs



Motivation für Secure Web-APIs



Die Gartner Group geht davon aus, dass im Jahr 2021 mehr als 60% aller Anwendungsentwicklungen unter Einsatz von Web-APIs erfolgen. Dies unterstreicht den Bedarf sich mit sicheren, vertrauenswürdigen und gesetzeskonformen Web-APIs auseinanderzusetzen!

Unter Berücksichtigung: Zumerle, D. et al. 2019. API Security. What You Need to Do to Protect Your APIs [online].
Verfügbar unter <https://www.gartner.com/en/documents/3956746/api-security-what-you-need-to-do-to-protect-your-apis>

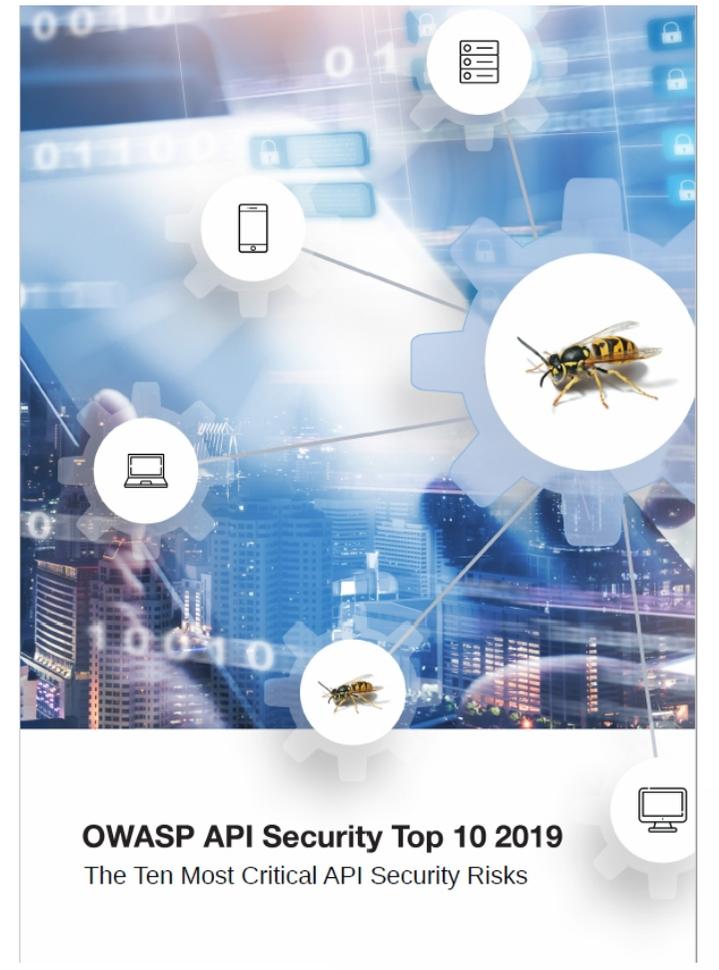


OWASP API Security Top 10



- API1:2019 Broken Object Level Authorization
- API2:2019 Broken User Authentication
- API3:2019 Excessive Data Exposure
- API4:2019 Lack of Resources & Rate Limiting
- ...

Quelle der rechten Abbildung: <https://owasp.org/www-project-api-security>,
Abruf: 30. März 2020





Pragmatische Industriesicht



- Einsatz von Tokens
 - Vertrauenswürdige Identitäten
 - Zugriffskontrolle auf Services und Ressourcen
- Einsatz von Verschlüsselung und Signaturen
 - Verschlüsselung mit Hilfe von TLS (Transport Layer Security)
 - Nur autorisierte Nutzer können Daten entschlüsseln und modifizieren.
- Identifizierung von Schwachstellen
 - Einsatz aktueller Versionen für Hard- und Software
 - Sniffer-Lösungen zur Erkennung von Sicherheitsproblemen
 - Berücksichtigung von „Zero-Day-Exploits“
- Einsatz von Quotas und Throttling
 - Überwachung der Aufrufquantität
 - Regeln zur Erkennung von Spikes und Denial-of-Service-Attacken
- Einsatz eines API Gateways

In Anlehnung an: <https://www.redhat.com/de/topics/security/api-security>



Online Tutorial der Brightcove University



Übersetzen
ÜBERSETZUNGEN VON GOOGLE TRANSLATE ZUR VERFÜGUNG GESTELLT

SUPPORTANFRAGE | SYSTEMSTATUS | BRIGHTCOVE UNIVERSITY



STARTSEITE | PRODUKTE ▾ | QUICK LINKS ▾ | AUSBILDUNG ▾ | AKTUELLES

Dokumentation

Suche



Überblick: OAuth API v4

Auf dieser Seite

Video Cloud Version

Brightcove Player Version

(What's the difference?)

- › Überblick
- › Glossar der Begriffe
- › Basis-URL
- › Client-Anmeldedatenfluss
- › Allgemeine Autorisierung
- › Erhalten Sie Clientanmeldeinformationen
- › Geschäftstätigkeit
- › Mit OAuth arbeiten
- › Client-Beispiele und Bibliotheken
- › Verwandte Inhalte

- 🔗 Video Cloud
- 👤 API-Entwickler
- 📄 API-Übersichten, Beglaubigung, OAuth API

Der Brightcove Die Implementierung von OAuth2 ermöglicht es Ihnen, Zugriffstokens für verschiedene zu erhalten Brightcove APIs.

Überblick

BrightcoveDie Umsetzung besteht aus zwei Teilen:

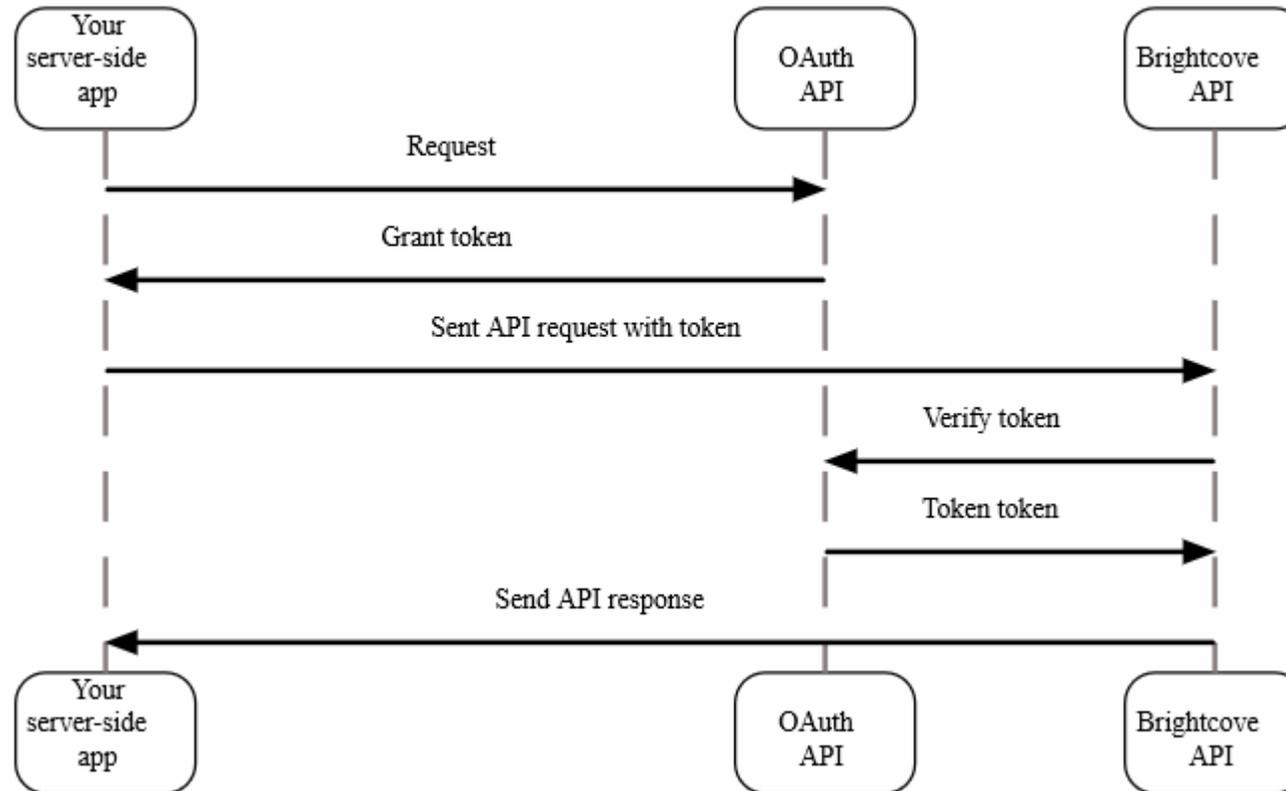
- Der OAuth API - Bietet Zugriff auf alle verfügbaren OAuth-Funktionen
- Die Benutzeroberfläche von OAuth Credentials - Über die Benutzeroberfläche für die Kontoeinstellungen in Studio kann die Benutzeroberfläche auf einfache Weise Apps registrieren, die verwendet werden Brightcove APIs und generieren eine Client-ID und ein Client-Secret für sie. Sehen [API-Authentifizierungsdaten verwalten](#) Anweisungen zur Verwendung der OAuth-Benutzeroberfläche

Siehe auch die [API-Referenz](#).

Quelle: <https://support.brightcove.com/de/overview-oauth-api-v4#Overview>,
Abruf: 30. März 2020



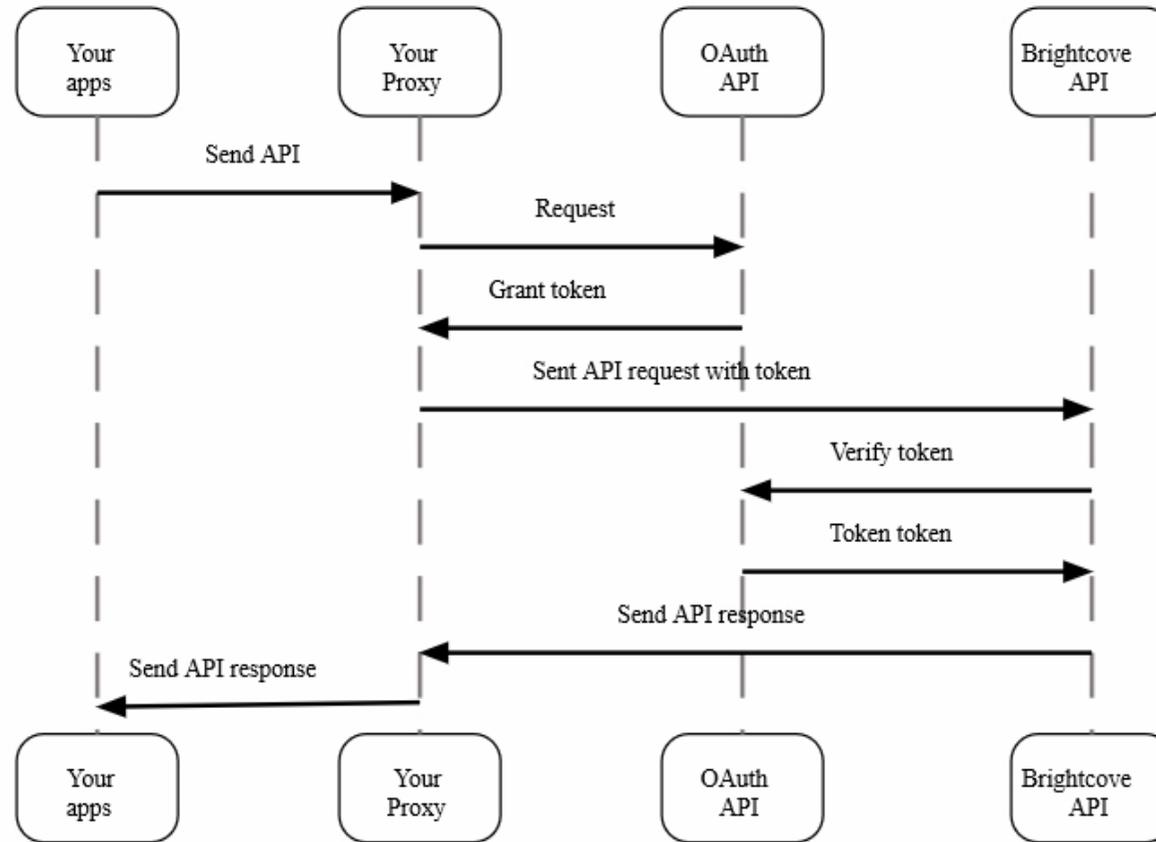
Interaktionsprinzip OAuth2



Quelle: <https://support.brightcove.com/de/overview-oauth-api-v4#Overview>,
Abruf: 30. März 2020



Interaktionsprinzip OAuth2 (via Proxy)



Quelle: <https://support.brightcove.com/de/overview-oauth-api-v4#Overview>,
Abruf: 30. März 2020



Literaturhinweis

- Grundlagen - Begriffe und Stakeholder.
- Analyse existierender Arbeiten zur API-Sicherheit.
- Sicherheit und Compliance im API-Management (Prozesse/Aufgaben).
- Qualitätssicherung – Secure Web-APIs
 - Konstruktiv
 - Analytisch
 - Betrieblich
- Innovative Lösungsansätze
 - DLT/Blockchain
 - Risiko- und Reifebewertungen
 - KI-basierte Ansätze





GraphQL



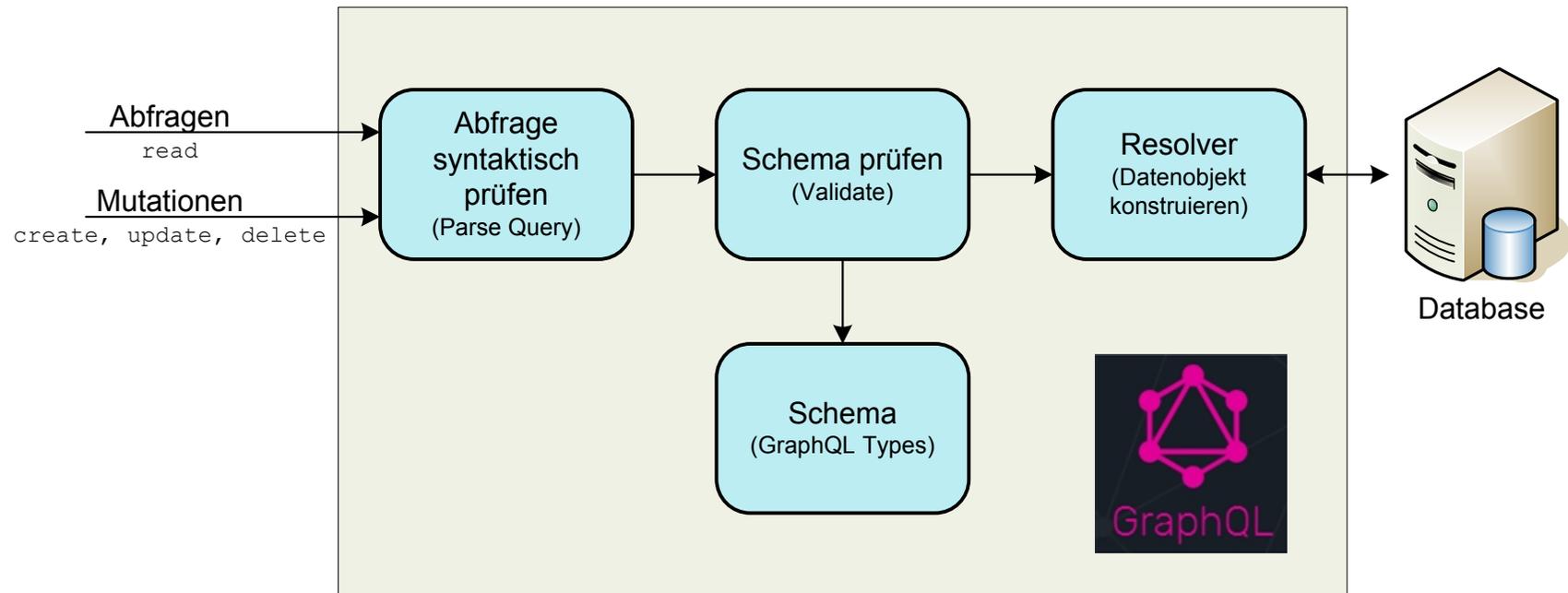
Motivation und Ziele GraphQL



- Ursprünglich in 2012 von Facebook veröffentlicht.
- Seit 2015 als Open Source bereitgestellt → <https://graphql.org>
- Adressierung des Problem des „Overfetching“.
- Adressierung des Problem feingranularer Ressourcenzugriffe.
- Abfragesprache (vgl. SQL) für webbasierte API (meist HTTP).
 - Alternative/Ergänzung zu REST-basierten Web-APIs
 - Alternative/Ergänzung zu XML-basierten Web Services.
- GraphQL Schema als Beschreibung der API



GraphQL Laufzeitarchitektur



In Anlehnung: redhat: Was ist GraphQL, <https://www.redhat.com/de/topics/api/what-is-graphql>, Abruf: Mai 2020

Quelle des GraphQL-Symbol: <https://graphql.org>, Abruf: Mai 2020



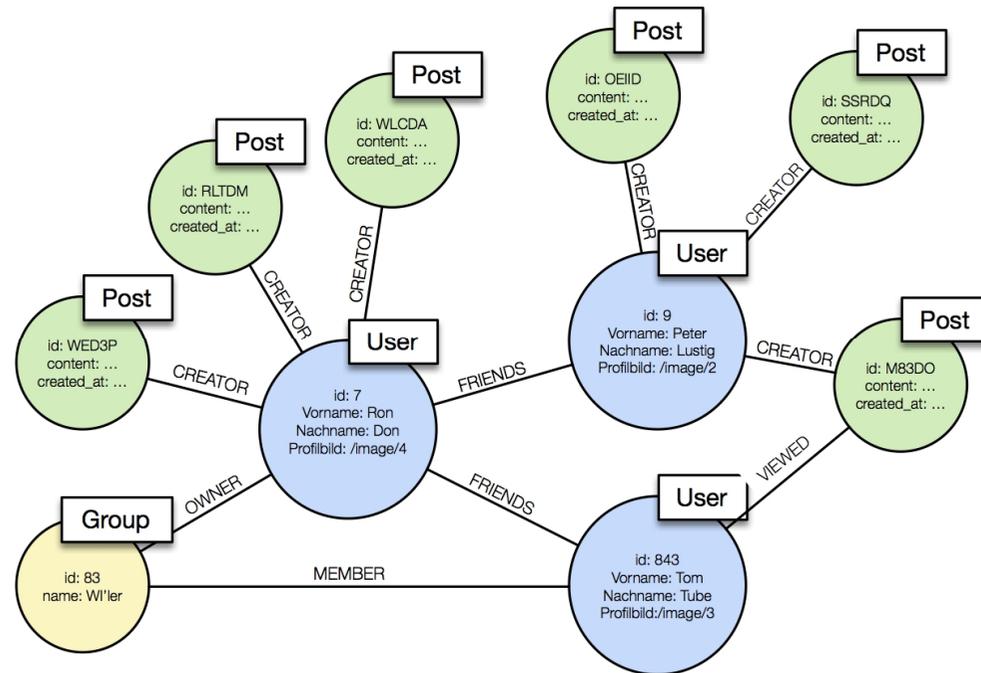
GraphQL Type-System (Schema)



```
type User {  
  id: Int  
  vorname: String  
  nachname: String  
  friends: [User]  
  groups: [Group]  
}
```

```
type Post {  
  id: Int  
  content: String  
  creator: User  
}
```

```
type Group {  
  id: Int  
  name: String  
  member: [User]  
}
```



Quelle: Böhme, L.; Grunert, J.: Einführung und Implementierung von GraphQL als API-Spezifikation, in Proc. Schmietendorf, A.; Nitze, A. (Hrsg.): API-First/API-Management Workshop ECC 2020, Shaker 2020



GraphQL Beispielabfrage

- Die Query wird durch die Verschachtelung von Type Eigenschaften erstellt.
- Das Beispiel zeigt eine Abfrage mit GraphQL, um den letzten Post vom User mit der ID 9 abzurufen.

```
query getPosts {  
  user(id: 9){  
    id  
    vorname  
    nachname  
    posts(last: 1){  
      id  
      content  
    }  
  }  
}
```

```
{  
  data: {  
    "id": 9,  
    "vorname": "Peter",  
    "nachname": "Lustig",  
    "posts": [  
      {  
        "id": "aW7nD",  
        "content": "ECC 2018"  
      }  
    ]  
  }  
}
```

Quelle: Böhme, L.; Grunert, J.: Einführung und Implementierung von GraphQL als API-Spezifikation, in Proc. Schmietendorf, A.; Nitze, A. (Hrsg.): API-First/API-Management Workshop ECC 2020, Shaker 2020



GraphQL Explorer unter GitHub



GitHub Developer Docs ▾ Blog Forum Versions ▾ Search...

GitHub GraphQL API Signed in as **schmiete**. You're ready to explore! Sign out

Heads up! GitHub's GraphQL Explorer makes use of your **real, live, production data**.

Explorer × GraphQL ▶ Prettify History Explorer < Docs

```
1 {
2   viewer {
3     login
4     projectsUrl
5     status {
6       createdAt
7     }
8   }
9   user(login: "") {
10    gistComments(after: "") {
11      edges {
12        node {
13          id
14        }
15      }
16    }
17  }
18 }
```

```
{
  "data": {
    "viewer": {
      "login": "schmiete",
      "projectsUrl": "https://github.com/users/schmiete/projects",
      "status": null
    },
    "user": null
  },
  "errors": [
    {
      "type": "NOT_FOUND"
    }
  ]
}
```

Quelle der Abb.: <https://developer.github.com/v4/explorer>, letzter Abruf: Mai 2020



Quellen zum Thema

- Kirschner, A.; Pointner, G.: Was der neue API-Ansatz leistet und was nicht - Einführung in GraphQL, <https://jaxenter.de/einfuehrung-in-graphql-71048>
- Klassen, D.; Augsten, S.: Eine andere Art APIs zu implementieren - GraphQL als Alternative zu REST, <https://www.dev-insider.de/graphql-als-alternative-zu-rest-a-752074>
- Springer, S.: GraphQL Server auf Basis von React entwickeln, <https://entwickler.de/online/development/graphql-server-react-579858102.html>



Kurzübung Themenkomplex 3



Kurzübung 3

(Alternative: Secure Web-APIs)

- Was wird unter dem Begriff “Zero-Day-Exploits” verstanden
- Machen Sie sich mit den “OWASP Top 10 API Security” Risiken vertraut
 - Gehen Sie für 2 Sicherheitslücken auf entwicklerseitige oder betriebliche Möglichkeiten zur Vermeidung bzw. Abmilderung ein.
 - Gehen Sie für 2 Sicherheitslücken auf die Möglichkeiten von Tests zur Aufdeckung potentieller Schwachstellen ein.
- Analysieren Sie die grundlegende Arbeitsweise von OAuth2.
 - Welche grundlegenden Rollen werden im OAuth2 unterschieden?
 - Welche Daten werden mittels Token bereitgestellt, wozu wird dieser genutzt?
 - Welche Zusammenhänge bestehen zwischen OAuth2 und OpenID?



Kurzübung 3

(Alternative: GraphQL)

- Machen Sie sich mit den grundlegenden Ideen und Prinzipien des GraphQL-Ansatzes vertraut.
- Welche Vor- und Nachteile sehen Sie beim Einsatz von GraphQL im Vergleich zu ausschließlich REST-basierten Web-APIs?
- Wie kann eine Serveranwendung um eine GraphQL-Schnittstelle erweitert werden? Gehen Sie auf die grundlegende Architektur und auf mindestens 2 Implementierungsalternativen (z.B. Framework Apollo) sowie die Schritte zur Entwicklung ein.
- Testen Sie eine fertig angebotene GraphQL-Schnittstelle (z.B. mit Hilfe des GitHub-GraphQL Explorers). → optionale Aufgabe!