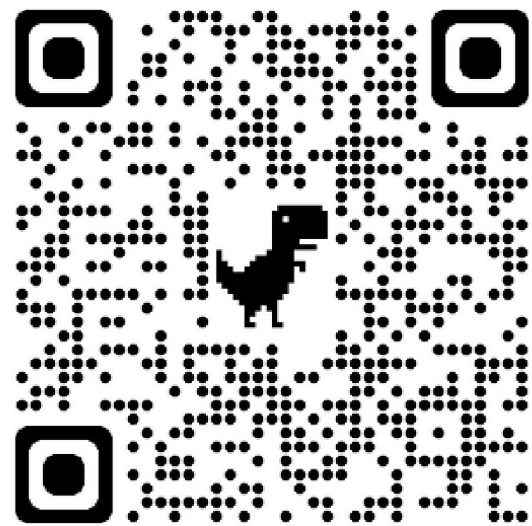




OAuth2 demystified

Presentation - 2024



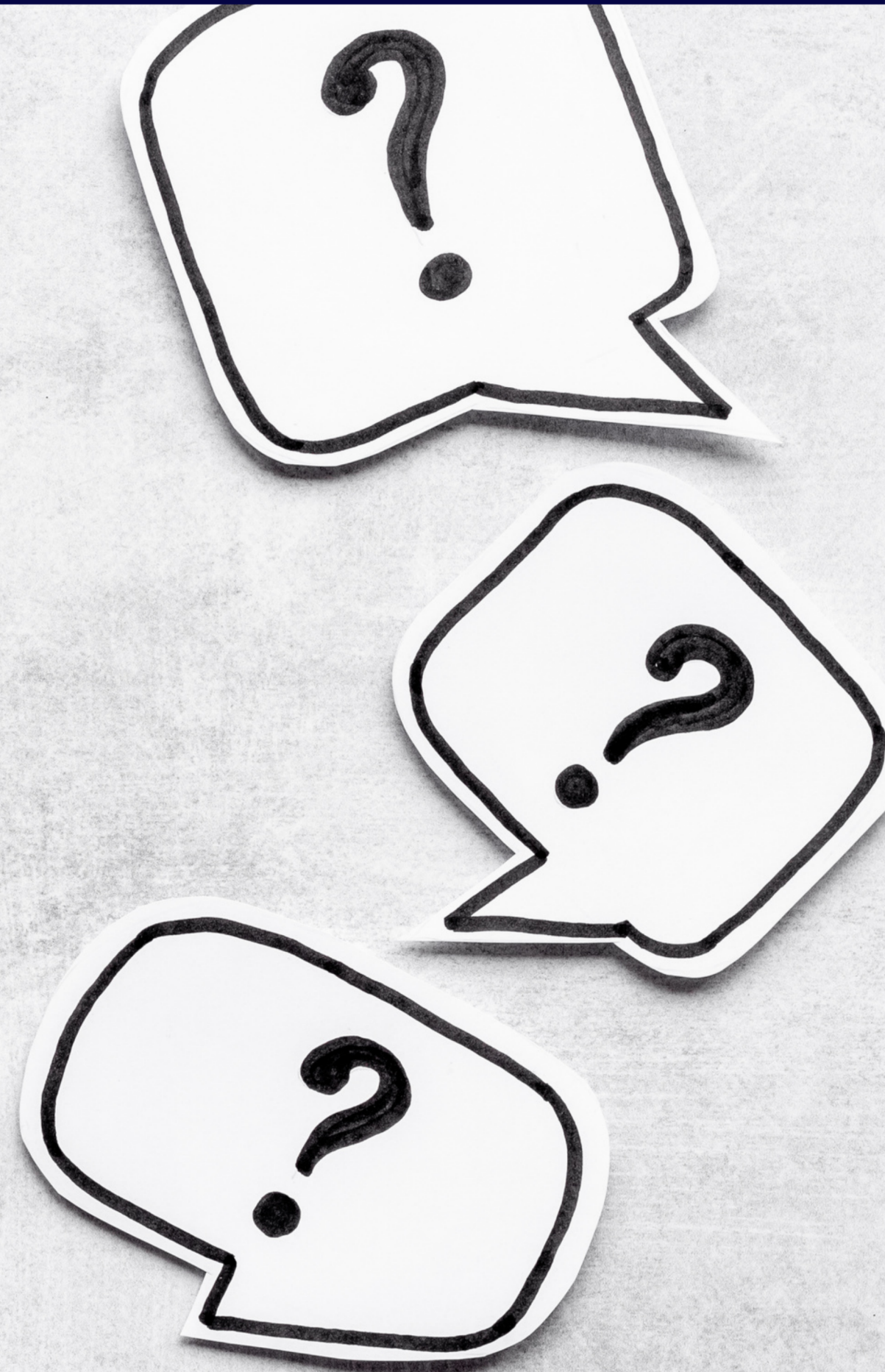
About Linda

- Freelance software developer. Bringing three decades of seasoned expertise in software development to the table. Find me at Lindalawton.dk
- Empowering businesses by seamlessly integrating cutting-edge Google technologies into their solutions as a freelance software developer.
- Recognized as a Google Developer Expert specializing in Google Analytics, Identity Platform, and AI/ML technologies.
- Daimto, a respected presence on Stack Overflow with an impressive reputation of over 110k points.



Objectives

- Understanding the difference between authentication and authorization.
- What OAuth 2.0 is and how it differs from basic login, and OAuth 1.0
- Describe the various OAuth 2.0 grant types and when it's appropriate to use each
- Implement an OAuth 2.0 server with curl



Authentication vs Authorization

"Who are you and what are you allowed to do?"



Authentication

- Verifying that someone is who they claim to be.
- Who are you?



Authorization

- Verifying which resources a user can access and what they are allowed to do with those resources.
- What are you allowed to do?

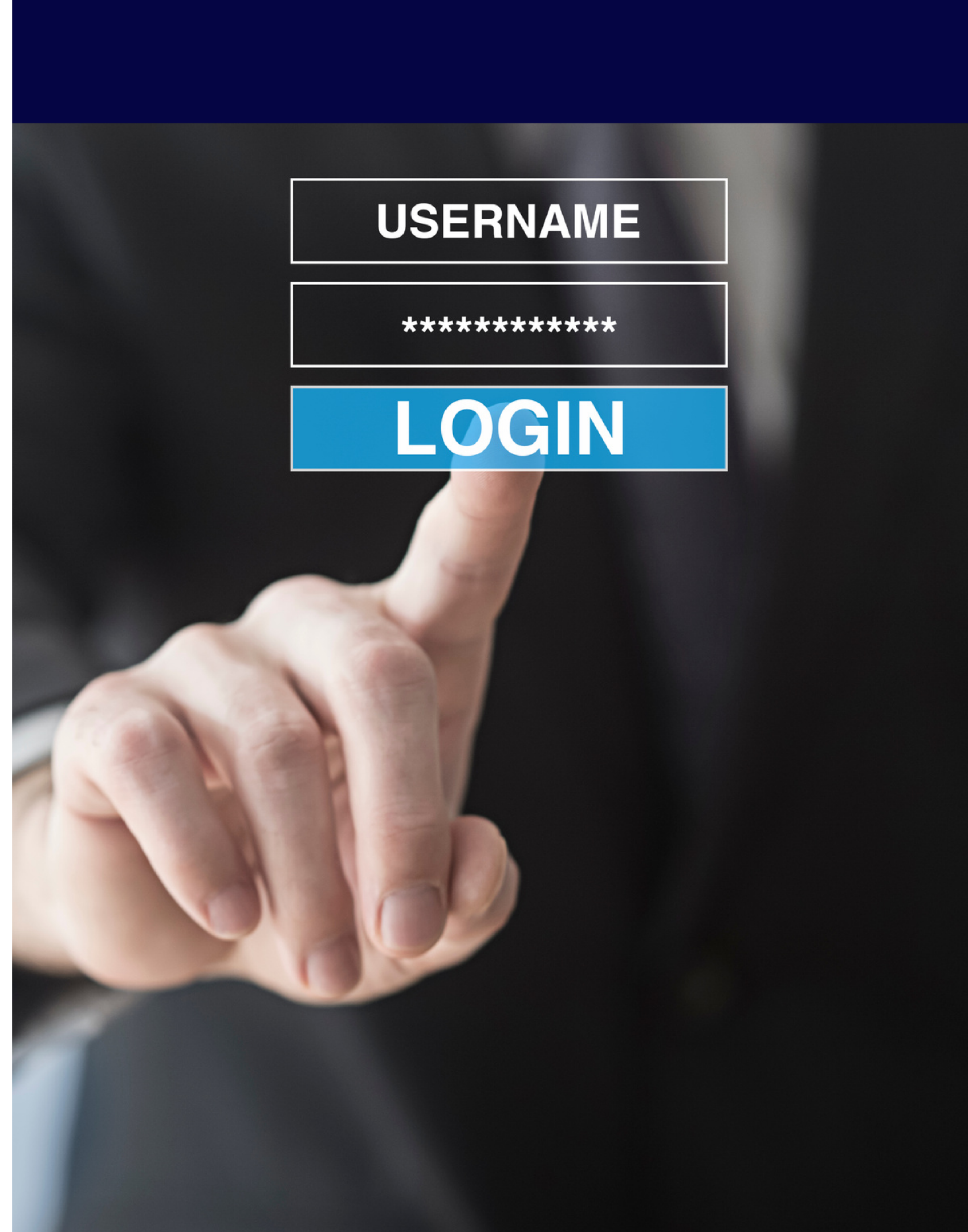
Authentication

Authentication refers to the process of verifying the identity of a user or entity.

It ensures that the user is who they claim to be.

Authentication mechanisms commonly involve providing credentials such as a username/password pair, biometric data (like fingerprints or facial recognition), security tokens, or digital certificates.

Once authenticated, the system grants the user access to its resources and functionalities.





Authorization

Authorization, on the other hand, comes after authentication and determines what actions or resources a user is permitted to access.

It involves enforcing rules and policies that govern what authenticated users are allowed to do within the system.

Authorization mechanisms typically involve assigning roles, permissions, or privileges to users or groups, specifying what they can or cannot access and what operations they can perform.

What is the difference between **Authentication and Authorization**



Summary

Authentication verifies the identity of users, while authorization controls what actions or resources they are allowed to access once authenticated. Both are crucial components of ensuring the security and integrity of software systems.



Introduction

What is OAuth?

OAuth (Open Authorization) is an open standard protocol that allows secure authorization of user access to resources without sharing their credentials.

It is widely used for enabling third-party applications to access web resources on behalf of a user.

OAuth works by allowing users to grant permissions to third-party applications to access their resources (such as data or services) hosted by another service provider (such as a social media platform or an online service). This is achieved through the exchange of tokens instead of sharing the user's credentials (like username and password).



Terminology

Term	Definition
Resource Owner (aka User)	End user who authorizes an application to access their account
Client (aka Consumer)	Application that accesses protected resources on behalf of the user
Authorization Server	Server which grants Access Tokens after the user authorizes the application
Resource Server	Restricted resource / API

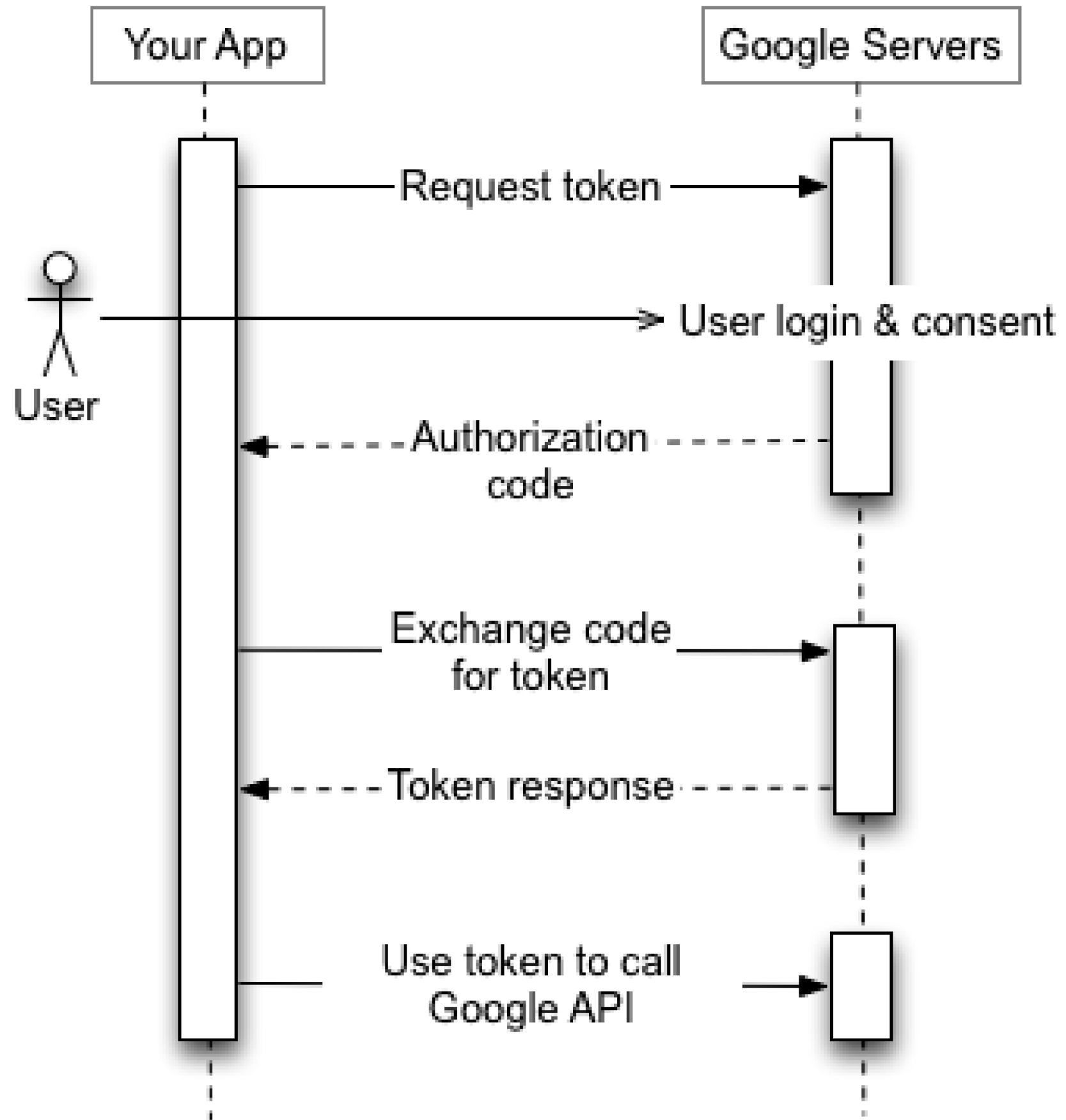
Terminology

Term	Definition
Access Token	Token used to access restricted resources
Authorization Code	Intermediate token returned to the client, after the user authorization step, which the client exchanges it for an Access Token
Refresh token	long lived token

Grant Types

Term	Definition
Authorization Code	Server-side web apps
Implicit	Single Page Apps (SPAs)
Client Credentials	Machine-to-machine

Understanding OAuth2



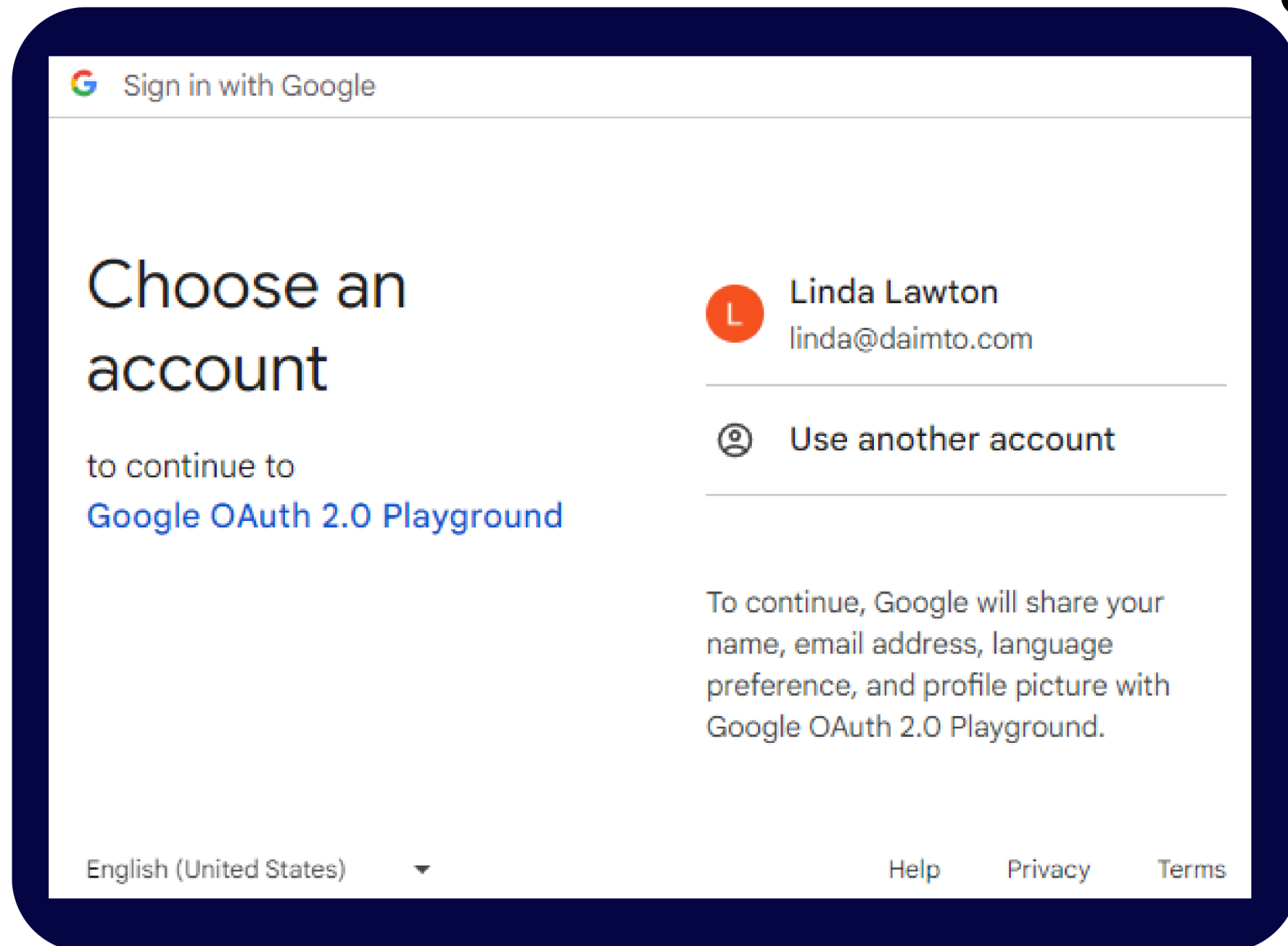
OAuth is a dance between the user, your application, and the authorization server.

When the steps of this dance have been preformed in the proper order

The application will have access to the users data.

If any of those steps are preformed incorrectly or out of order. An error will occur.

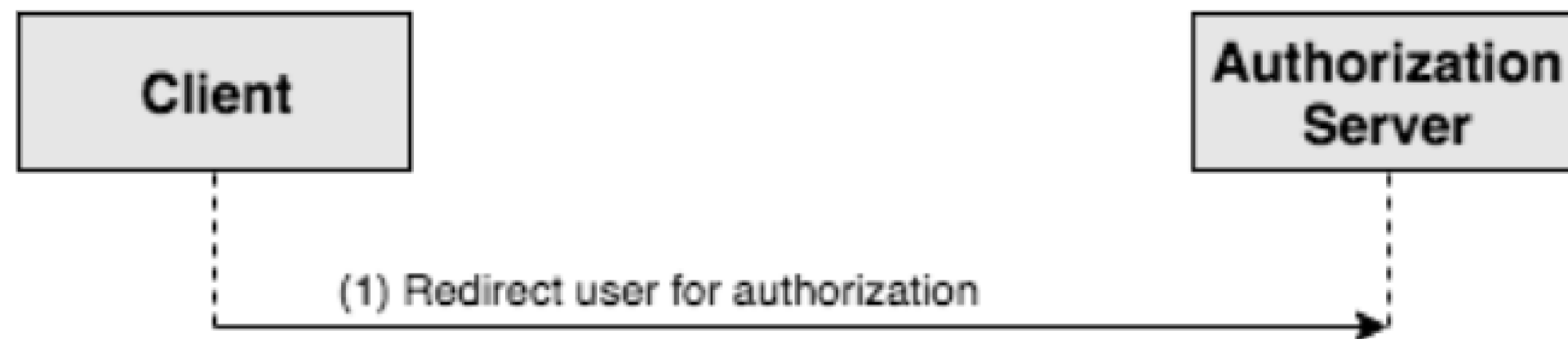
Sign-in




```
https://accounts.google.com/o/oauth2/v2/auth/  
oauthchooseaccount?  
redirect_uri=https%3A%2F%2Fdevelopers.google.c  
om%2Foauthplayground&prompt=consent&respo  
nse_type=code&client_id=  
[REDACTED]&scope=profile&access_type=offline
```


Authorization Code Flow


- [Step 1] Client redirects the Resource Owner to the Authorization Server





 Sign in with Google



Google APIs Explorer wants to access your Google Account

 support@daimto.com

 This app is intended to help you explore Google's API offerings. Do not share your API Credentials with anyone, and do not copy data from your account into apps that you do not trust.

The app's access to your account will be revoked after 7 days. You can also remove access sooner in your [Google Account](#).

 See and download all your Google Drive files 

 View and manage metadata of files in your Google Drive 

Make sure you trust Google APIs Explorer

You may be sharing sensitive info with this site or app. Learn about how Google APIs Explorer will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your [Google Account](#).

[Learn about the risks](#)


Cancel

Allow

 Sign in with Google

Google APIs Explorer wants to access your Google Account

 support@daimto.com

 This app is intended to help you explore Google's API offerings. Do not share your API Credentials with anyone, and do not copy data from your account into apps that you do not trust.

The app's access to your account will be revoked after 7

Google Account.



See and download all your Google Drive files



View and manage metadata of files in your Google Drive



Make sure you trust Google APIs Explorer



See and download all your Google Drive files



View and manage metadata of files in your Google Drive



Make sure you trust Google APIs Explorer

You may be sharing sensitive info with this site or app. Learn about how Google APIs Explorer will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your [Google Account](#).

[Learn about the risks](#)

Cancel

Allow

English (United States)



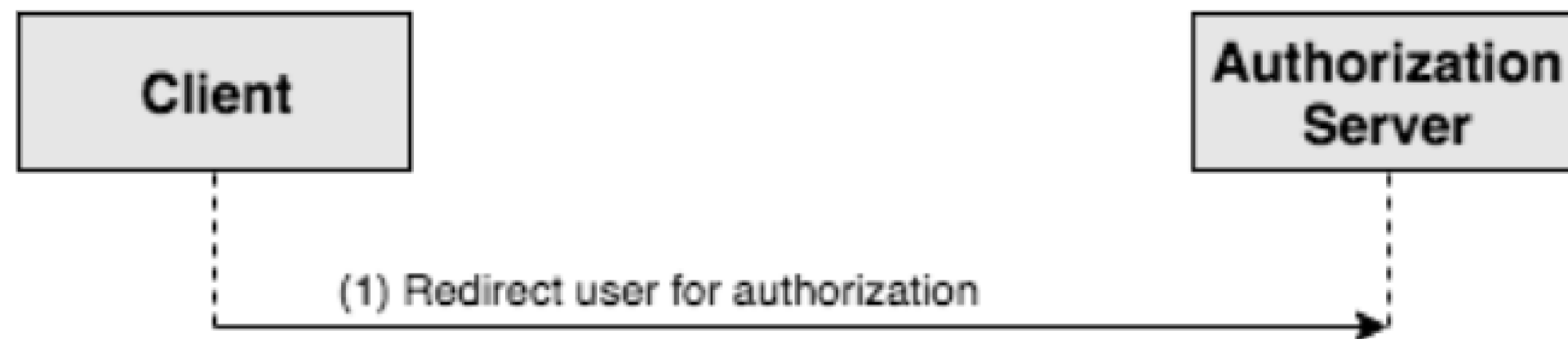
[Help](#)

[Privacy](#)


[Terms](#)

Authorization Code Flow


- [Step 1] Client redirects the Resource Owner to the Authorization Server




- [Step 2] Resource Owner grants authorization



 Sign in with Google



Google APIs Explorer wants to access your Google Account

 support@daimto.com

 This app is intended to help you explore Google's API offerings. Do not share your API Credentials with anyone, and do not copy data from your account into apps that you do not trust.

The app's access to your account will be revoked after 7 days. You can also remove access sooner in your [Google Account](#).

 See and download all your Google Drive files 

 View and manage metadata of files in your Google Drive 

Make sure you trust Google APIs Explorer

You may be sharing sensitive info with this site or app. Learn about how Google APIs Explorer will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your [Google Account](#).

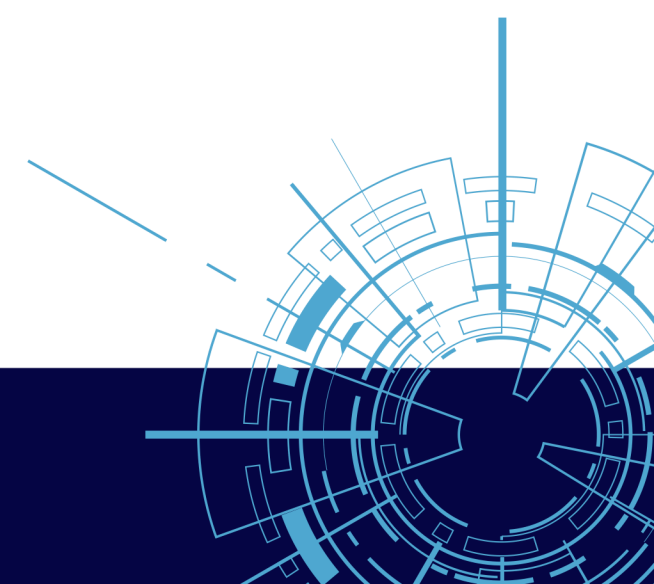
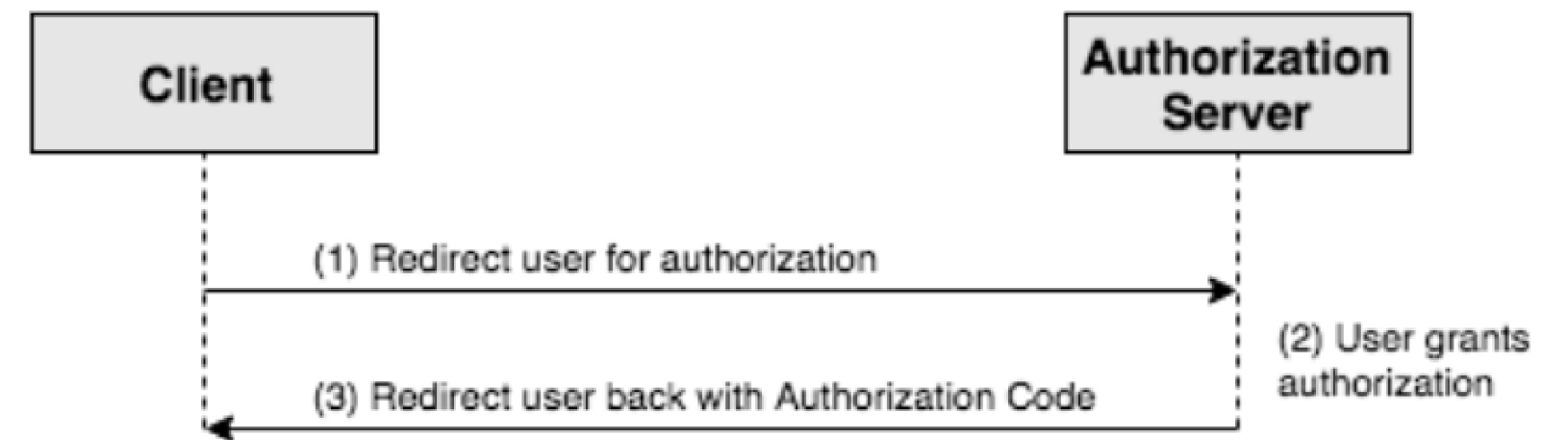
[Learn about the risks](#)

Cancel

Allow

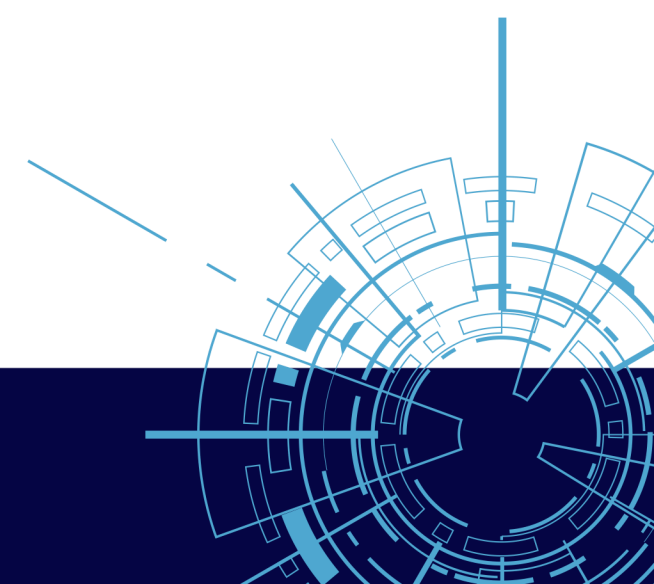
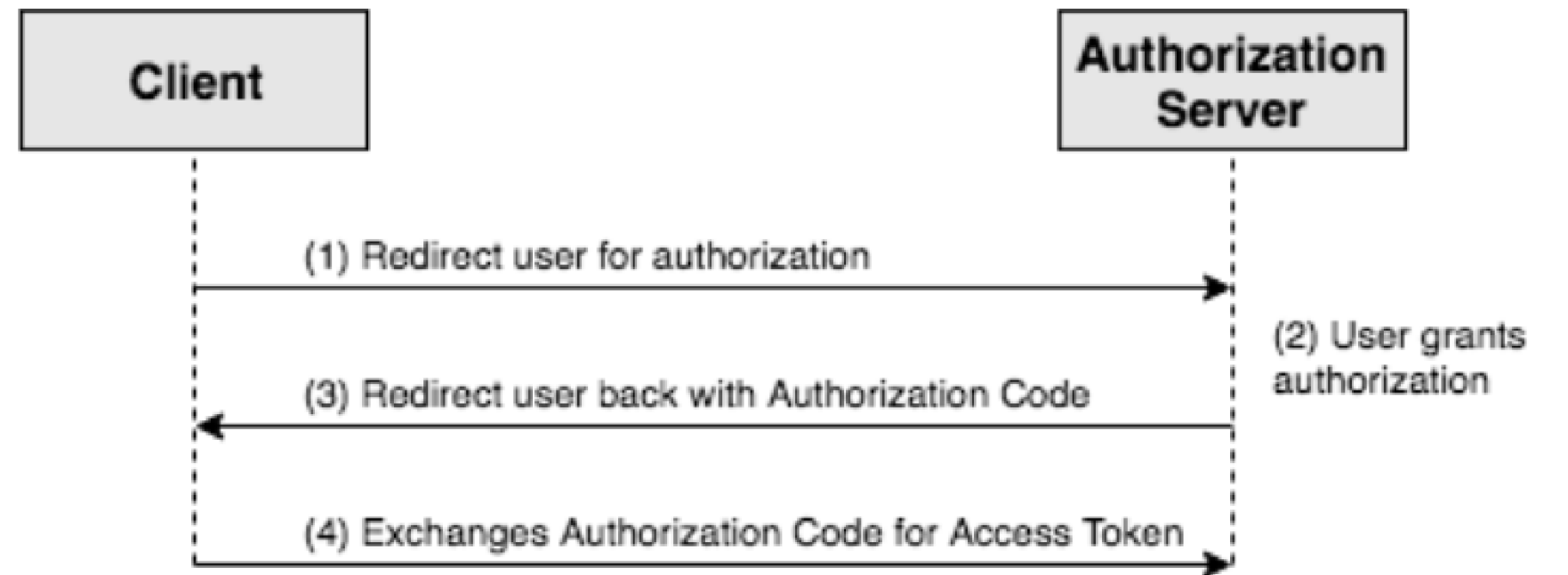
Exchange Authorization code

[Step 3] Authorization Server redirects the Resource Owner back to the Client with an Authorization Code



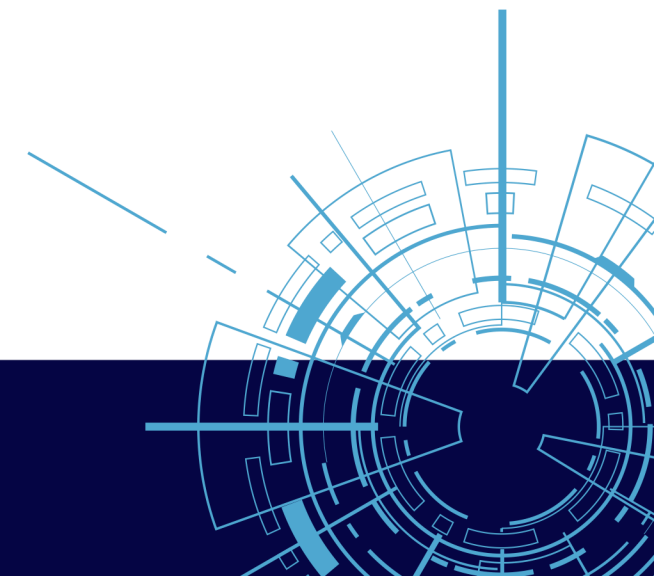
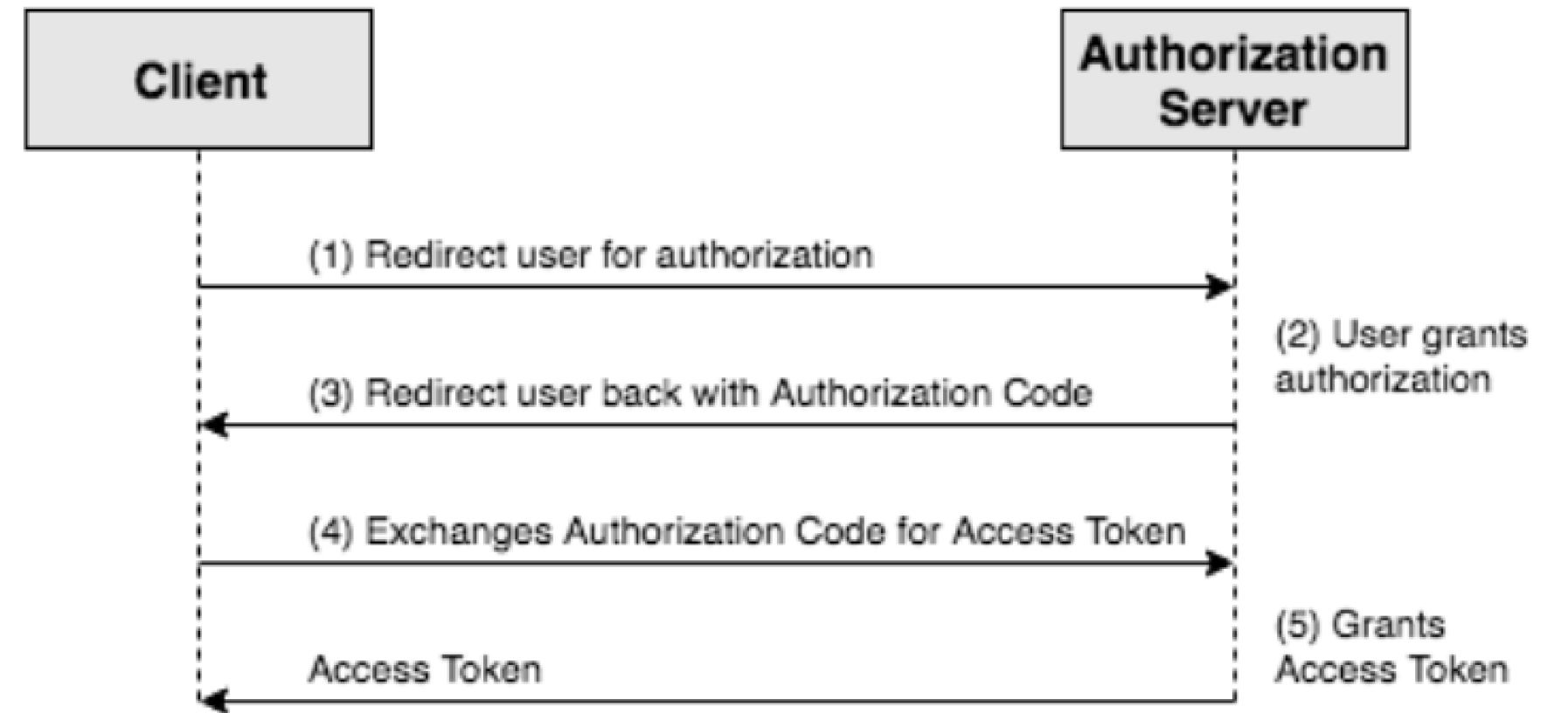
Authorization Code Flow (continued...)

[Step 4] Client exchanges Authorization Code for Access Token



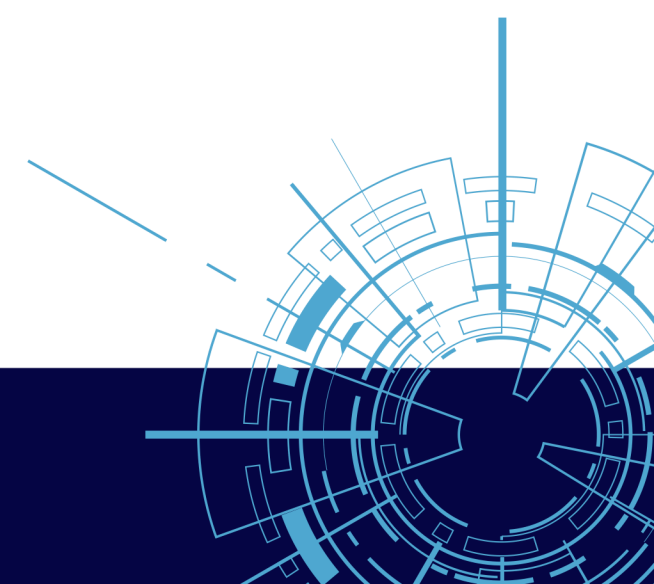
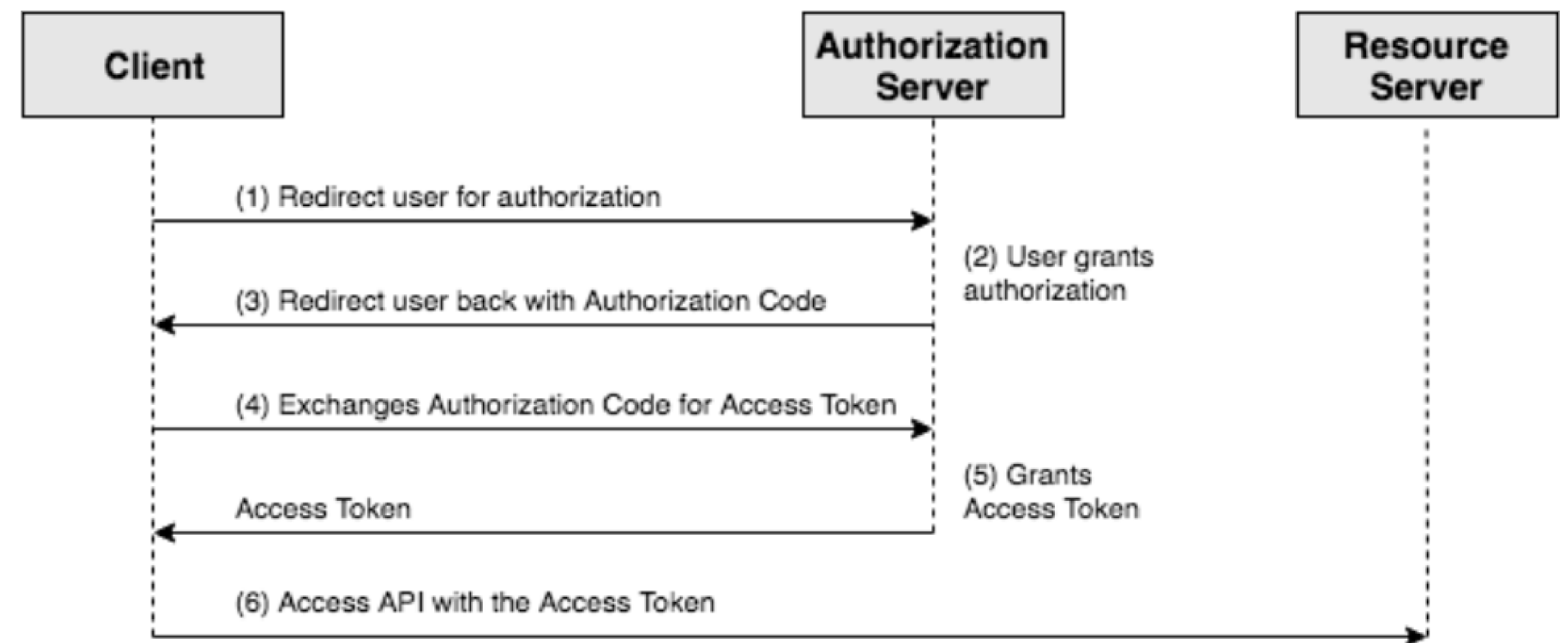
Authorization Code Flow (continued...)

[Step 5] Authorization Server grants Access Token



Authorization Code Flow (continued...)

[Step 6] Client uses the Access Token to access protected resources, from the Resource Server, on behalf of the Resource Owner



Why is this important?

Before OAuth

The User would provide their credentials directly to the Client app.

Problems

1. Client stores User's password (one more application with your password)
2. Client gets complete access to the User's account (scope)
3. User cannot revoke access to the Client unless they reset their password



Information

Basic Authentication

Involves sending the user's credentials directly to the API server with each request.

```
# Combine username and password with a colon separator
credentials = f"{username}:{password}"

# Encode the credentials in Base64
credentials_bytes = credentials.encode('utf-8')
base64_credentials = base64.b64encode(credentials_bytes).decode('utf-8')
```

```
GET /api/resource HTTP/1.1
Host: example.com
Authorization: Basic dXNlcjpwYXNzd29yZA==
```

Basic Authentication

Considered less secure for several reasons

- 01** No Encryption
- 02** No Protection Against Replay Attacks
- 03** No Token Expiration
- 04** Lack of Granular Access Control
- 05** Storage of Credentials



Information

OAuth 1 Authentication

OAuth 1.0 is an authentication protocol that allows users to grant third-party applications access to their resources without sharing their passwords directly

```
curl -X GET \  
  --url 'https://api.example.com/resource' \  
  --header 'Authorization: OAuth oauth_consumer_key="your_consumer_key",  
oauth_token="your_access_token", oauth_signature_method="HMAC-SHA1",  
oauth_timestamp="timestamp", oauth_nonce="nonce", oauth_version="1.0",  
oauth_signature="generated_signature"'
```


OAuth 1 Authentication

Better than Basic auth.

- 01 Token-Based Authentication
- 02 Signature-Based Security
- 03 Three-Legged OAuth Flow
- 04 Complexity
- 05 Storage of Credentials

Its not all good

- 01 Complexity
- 02 Token Management
- 03 Limited Browser Support
- 04 Scalability
- 05 Token Expiration and Revocation



Information

OAuth 2 Authentication

OAuth 2.0 request authorization returned is an authorization code.

Request / Response

```
HTTP/1.1 302 Found
Location: https://accounts.google.com/o/oauth2/v2/auth?
redirect_uri=https%3A%2F%2Fdevelopers.google.com%2Foauthplayground&prompt=consent&response_type=code&client_id=407408718192.
apps.googleusercontent.com&scope=profile&access_type=offline
```

```
GET /oauthplayground/?code=4%2F0AeaYSHCiAtW-
wZfysPRMBzxaMSR7wwcUGa4LJw4X3w5S4F9xHjVUmi5oYJu_mZiC70EFHw&scope=profile+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.
profile HTTP/1.1
Host: developers.google.com
```



Information

Oauth 2 Authentication

Exchange the authorization code.

Request / Response

```
POST /token HTTP/1.1
Host: oauth2.googleapis.com
Content-length: 261
content-type: application/x-www-form-urlencoded
user-agent: google-oauth-playground
```

```
code=4%2F0AeaYSHCiAtW-wZfysPRMBzxaMSR7wwcUGa4Ljw4X3w5S4F9xHjVUmi5oYJu_mZiC7OEFHw&redirect_uri=https%3A%2F%2Fdevelopers.google.com%2Foauthplayground&client_id=407408718192.apps.googleusercontent.com&client_secret=*****&scope=&grant_type=authorization_code
```


OAuth 2 Authentication

[illegible]

OAuth 2 Authentication

Better than Basic auth.

01 Token-Based Authentication

02 Signature-Based Security

03 Three-Legged OAuth Flow

04 Complexity

05 Storage of Credentials

Its not all good

01 Complexity

02 Token Management

03 Limited Browser Support

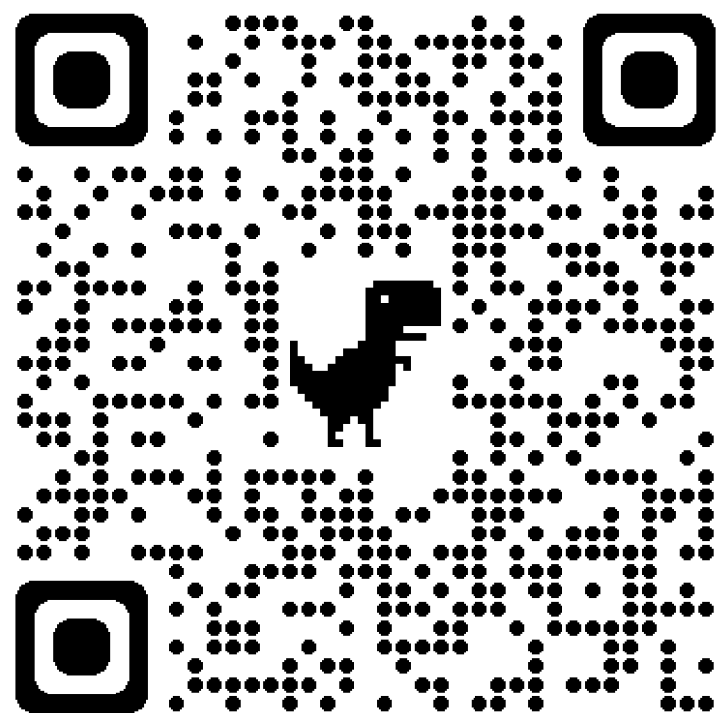
04 Scalability

05 Token Expiration and Revocation



Thank You

Freelance contact me



+45 22879566



Linda@daimto.com



LindaLawton.dk

