SSO Protocols
OAuth, OpenID Connect

Tasos Papadopoulos
Foundations of Cryptography
Aristotle University of Thessaloniki

Contents

- Authentication vs Authorization
- Need for OAuth protocol
- Delegated authorization with OAuth 2.0
- OAuth 2.0 flows
- Need for OpenID Connect
- OpenID Connect flow

Authentication vs Authorization

Authentication confirms that users are who they say they are. Authorization gives those users permission to access a protected resource.

In secure environments, authorization must always follow authentication. Users should first prove that their identities are genuine before an organization’s administrators grant them access to the requested resources.

Need for OAuth protocol

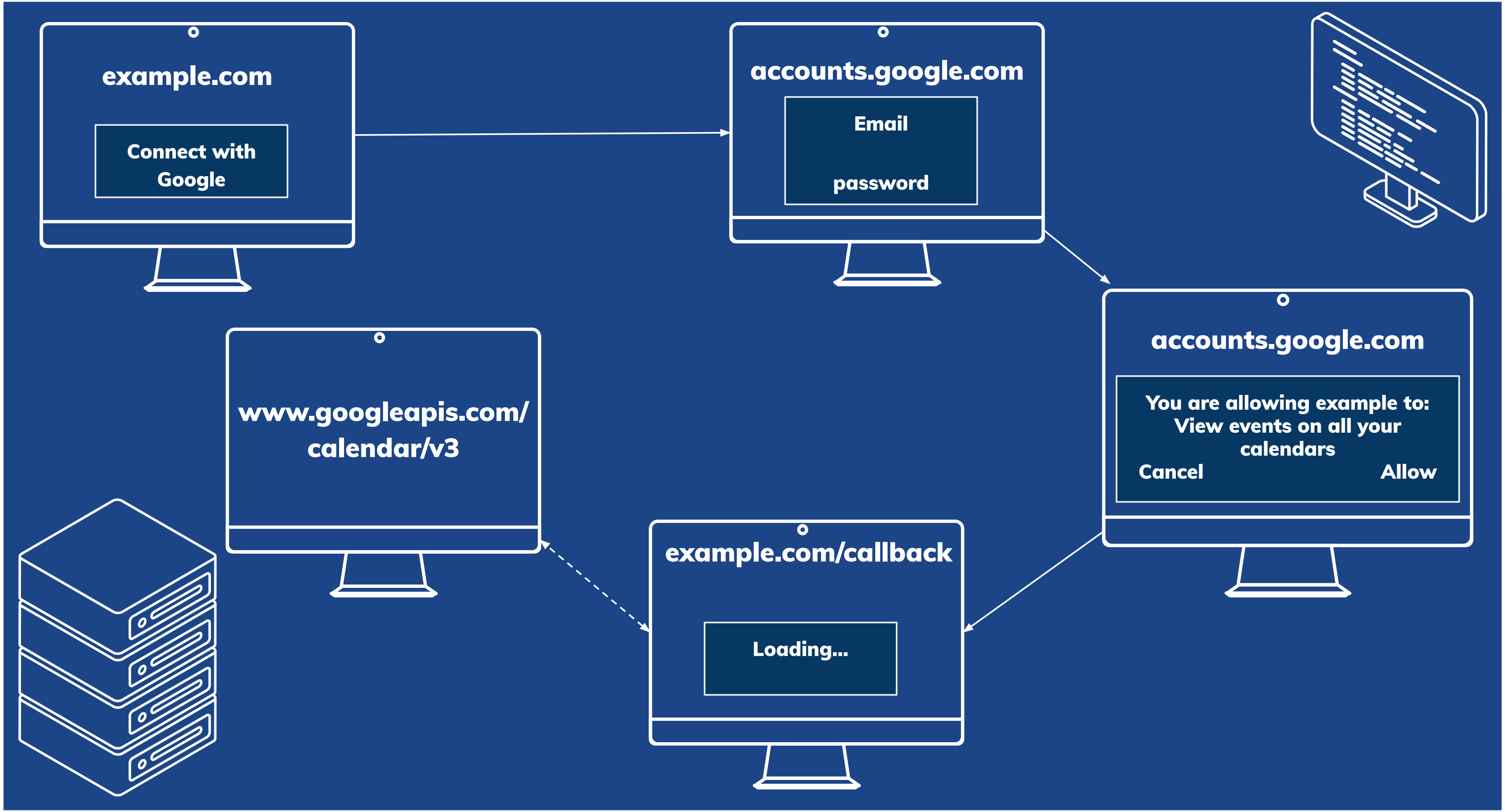
before **2007** **2012** **2018** **today**

Users login using forms and cookies for SSO using the SAML protocol for example. There is no support/need for mobile app login(iPhone launched on January 9, 2007) and delegated authorization

Applications use OAuth 2.0 both for Authentication and Authorization resulting in various implementations for getting user’s information and no common set of scopes

Many sites and large companies integrated OpenID Connect support and use it for use cases such as simple login, SSO and mobile app login and no longer for delegated authorization (where OAuth 2.0 is used instead as this was the use case that it was created for)

Delegated authorization with OAuth 2.0



example.com

Connect with Google

accounts.google.com

Email

password

accounts.google.com

You are allowing example to:
View events on all your calendars

Cancel

Allow

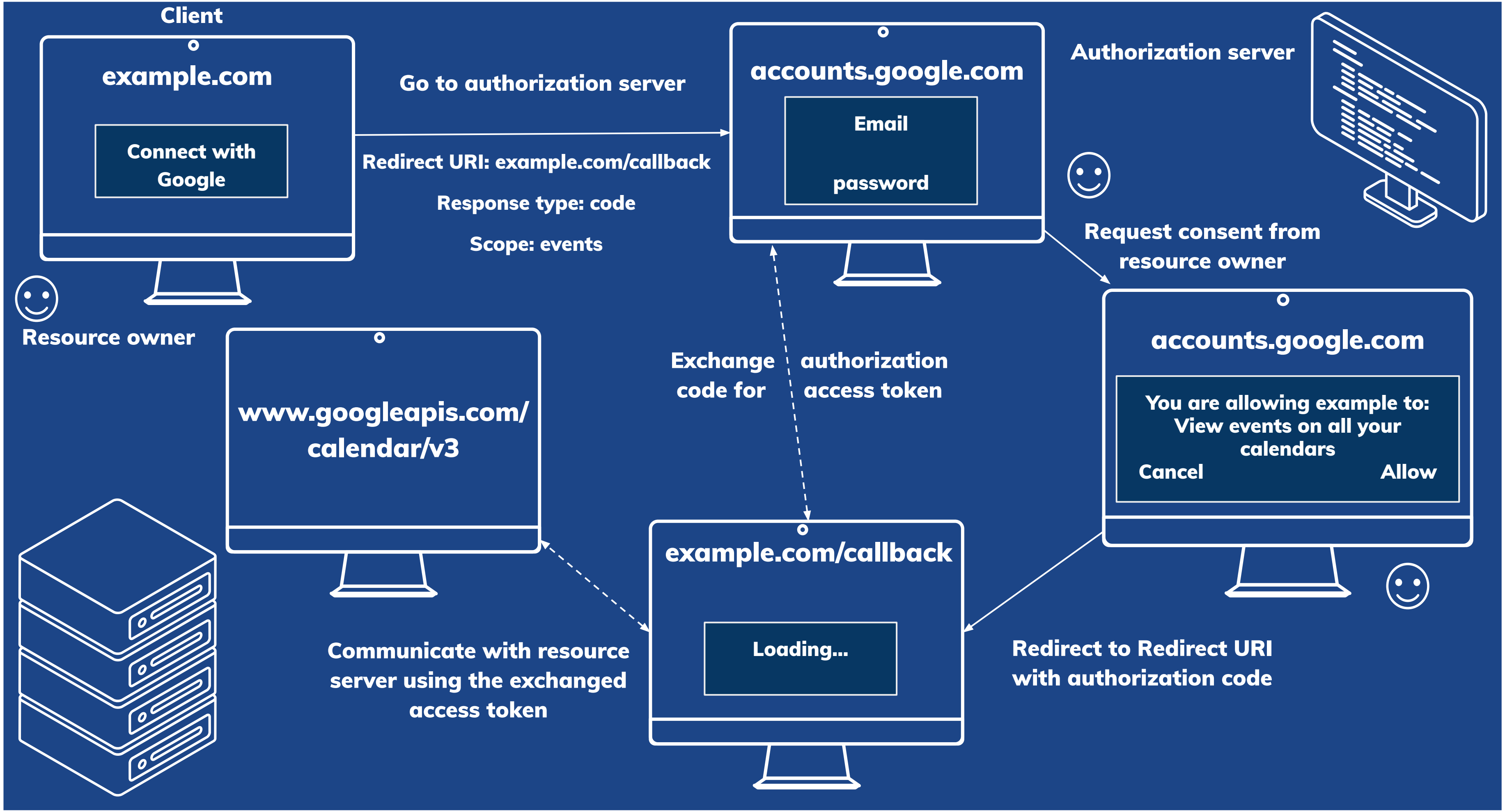
example.com/callback

Loading...

www.googleapis.com/calendar/v3

OAuth 2.0 terminology

- Resource owner
- Client
- Authorization server
- Resource server
- Authorization grant
- Redirect URI
- Access token
- Scope
- Consent



Client

example.com

Connect with Google

Resource owner

Go to authorization server

Redirect URI: example.com/callback

Response type: code

Scope: events

accounts.google.com

Email

password

Authorization server

Request consent from resource owner

accounts.google.com

You are allowing example to:
View events on all your calendars

Cancel Allow

Redirect to Redirect URI with authorization code

Exchange authorization code for access token

example.com/callback

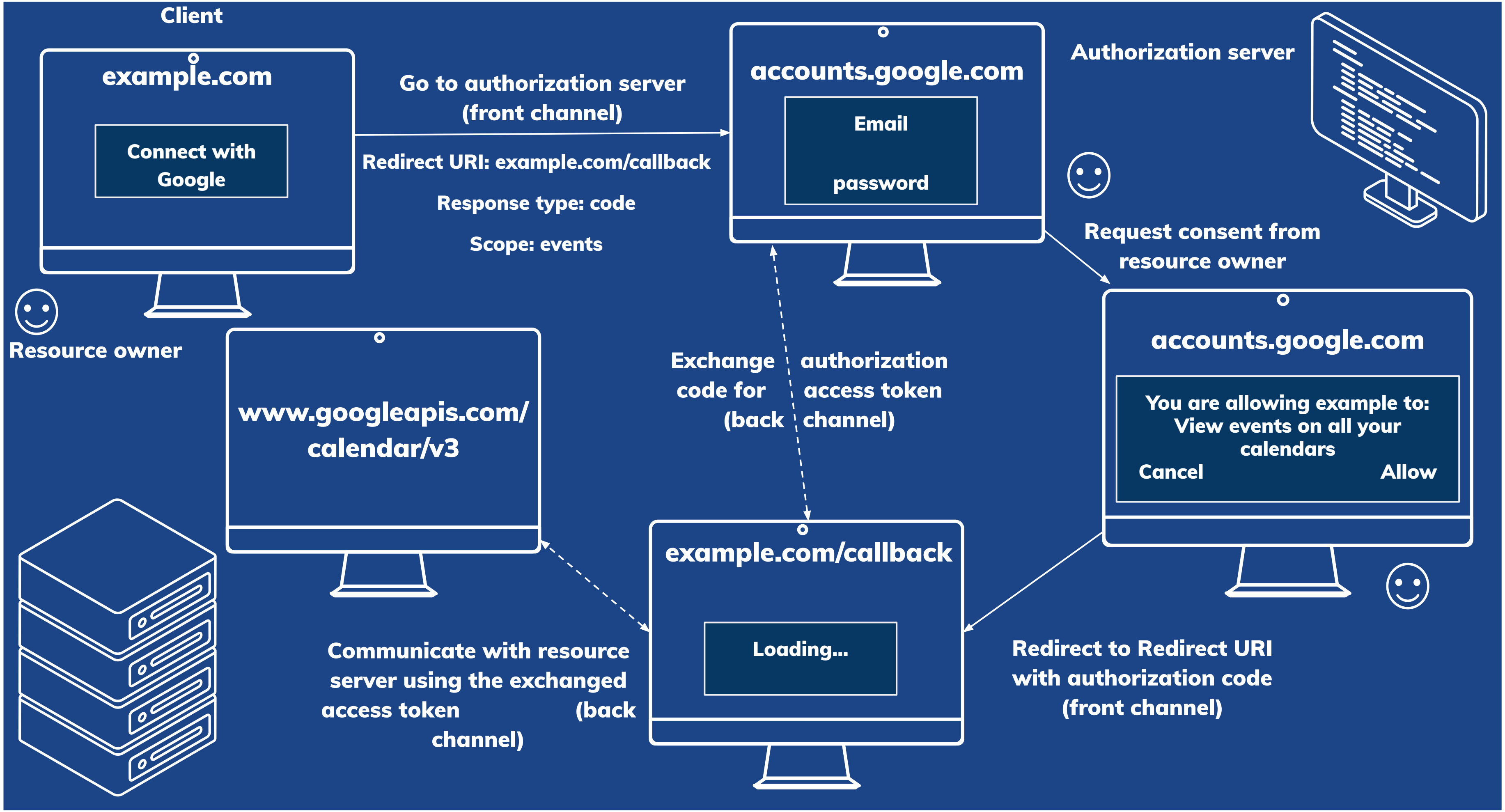
Loading...

www.googleapis.com/calendar/v3

Communicate with resource server using the exchanged access token

More terminology

- Back channel (highly secure channel, HTTPS between communication of backend services)
- Front channel (less secure channel)



Client

example.com

Connect with Google

Resource owner

Go to authorization server (front channel)

Redirect URI: example.com/callback

Response type: code

Scope: events

accounts.google.com

Email

password

Authorization server

Request consent from resource owner

accounts.google.com

You are allowing example to:
View events on all your calendars

Cancel Allow

Redirect to Redirect URI with authorization code (front channel)

example.com/callback

Loading...

Exchange authorization code for access token (back channel)

www.googleapis.com/calendar/v3

Communicate with resource server using the exchanged access token (back channel)

Authorization server returns an access token

```
{
    “access_token”: “abfabf1239817fcabab”,
    “expires_in”: 3600,
    “token_type”: “Bearer”,
}
```

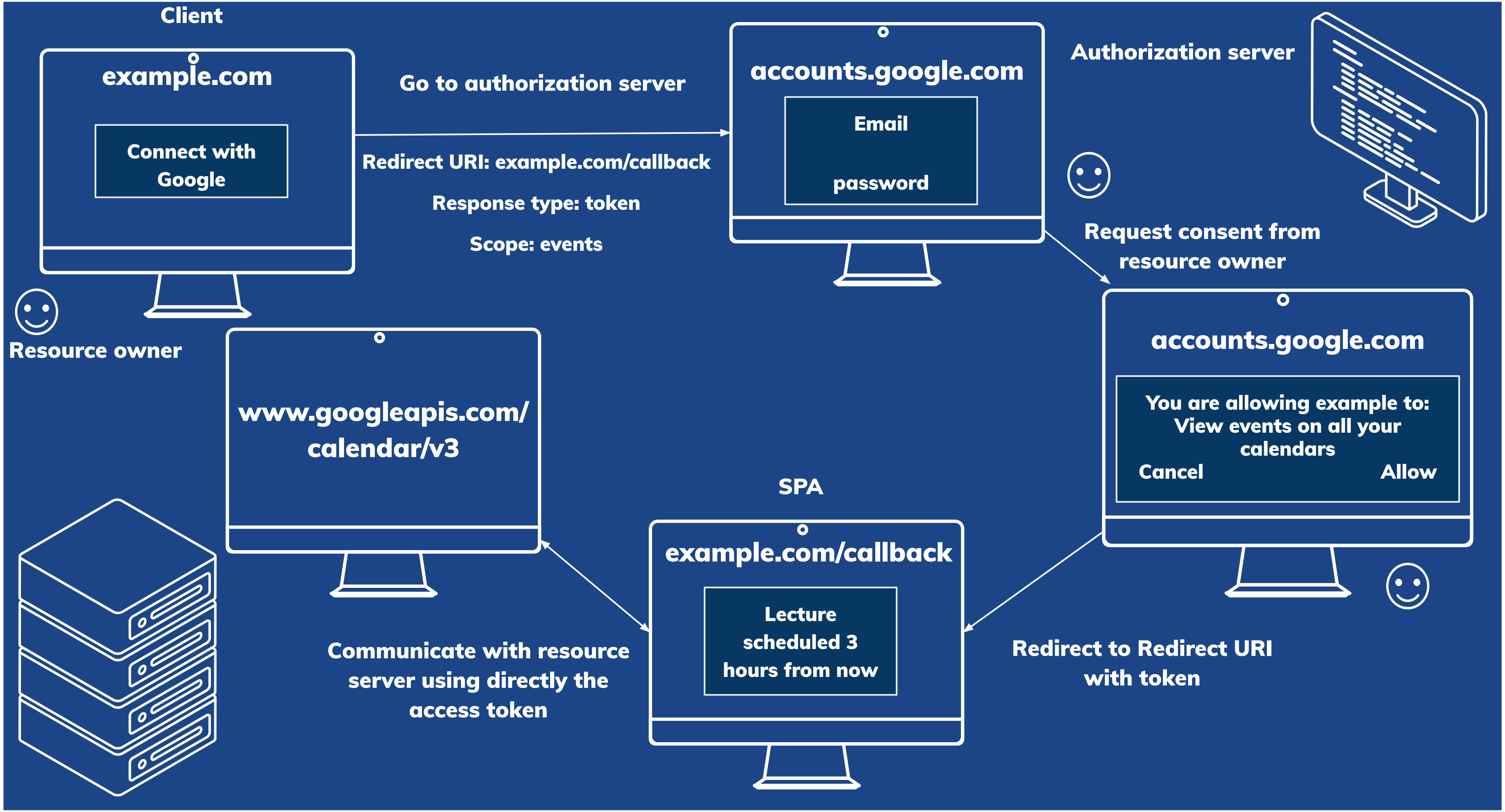
OAuth 2.0 flows

Authorization code

(front + back channel)

Implicit

(front channel only)



Client

example.com

Connect with Google

Go to authorization server

Redirect URI: example.com/callback

Response type: token

Scope: events

accounts.google.com

Email

password

Authorization server

Request consent from resource owner

accounts.google.com

You are allowing example to:
View events on all your calendars

Cancel Allow

Redirect to Redirect URI with token

SPA

example.com/callback

Lecture scheduled 3 hours from now

Communicate with resource server using directly the access token

www.googleapis.com/calendar/v3

Resource owner

before

2007

2012

2018

today

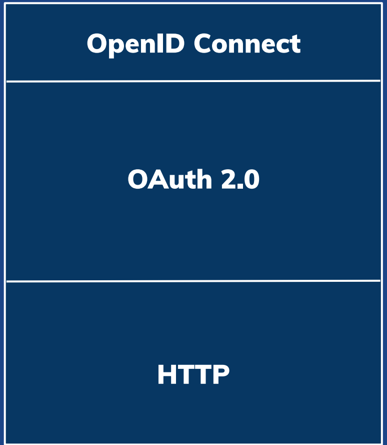
Users login using forms and cookies for SSO using for example the SAML protocol. There is no support/need for mobile app login(iPhone launched on January 9, 2007) and delegated authorization

Applications use OAuth 2.0 both for Authentication and Authorization resulting in various implementations for getting user's information and no common set of scopes

Many sites and large companies integrated OpenID Connect support and use it for use cases such as simple login, SSO and mobile app login and no longer for delegated authorization (where OAuth 2.0 is used instead as this was the use case that it was created for)

Need for OpenID Connect

Need for OpenID Connect



OpenID Connect

OAuth 2.0

HTTP

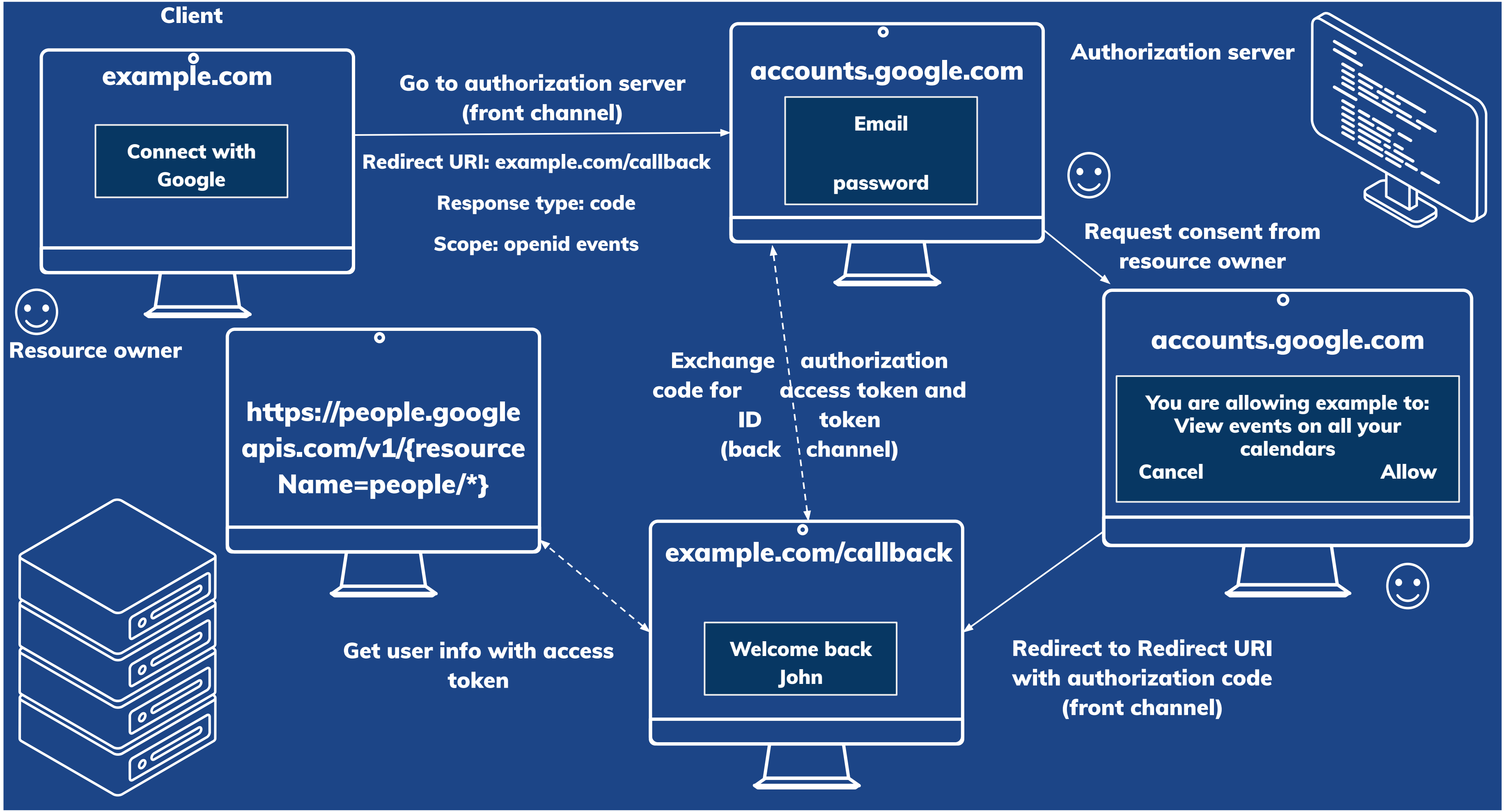
OpenID Connect for authentication

OAuth 2.0 for authorization

OpenID Connect adds

- ID token
- UserInfo endpoint for getting more user information
- Standard set of scopes
- Standardized implementation

OpenID Connect flow



Client

example.com

Connect with Google

Resource owner

Go to authorization server (front channel)

Redirect URI: example.com/callback

Response type: code

Scope: openid events

accounts.google.com

Email

password

Authorization server

Request consent from resource owner

accounts.google.com

You are allowing example to:
View events on all your calendars

Cancel

Allow

Redirect to Redirect URI with authorization code (front channel)

Exchange authorization code for access token and ID token (back channel)

example.com/callback

Welcome back John

Get user info with access token

https://people.googleapis.com/v1/{resourceName=people/*}

On OpenID Connect flow, authorization server returns access and ID tokens

```
{
    “access_token”: “abfabf1239817fcabab”,
    “id_token”: “abfa7fcabababfdabd”,
    “expires_in”: 3600,
    “token_type”: “Bearer”,
}
```

ID token (JWT)

JSON Web Token

eyJhbGciOiJIUzI1NiIsInR5c
CI6IkpXVCJ9

Header

.

eyJzdWIiOiIxMjM0NTY3O
DkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaWF0IjoxNTE2
MjM5MDIyfQ

Payload

.

SflKxwRJSMeKKF2QT4fw
pMeJf36POk6yJV_adQssw
5c

Signature

before

2007

2012

2018

today

Users login using forms and cookies for SSO using for example the SAML protocol. There is no support/need for mobile app login(iPhone launched on January 9, 2007) and delegated authorization

Applications use OAuth 2.0 both for Authentication and Authorization resulting in various implementations for getting user’s information and no common set of scopes

Many sites and large companies integrated OpenID Connect support and use it for use cases such as simple login, SSO and mobile app login and no longer for delegated authorization (where OAuth 2.0 is used instead as this was the use case that it was created for)

Thank you!

Questions?

Fonts Used: Muli and Source Code Pro
Presentation Template: SlidesMania
Images: Unsplash

Tasos Papadopoulos
apapadoi@csd.auth.gr

References

- https://www.oauth.com/oauth2-servers/map-oauth-2-0-specs/
- https://github.com/iamshaunjp/oauth-playlist
- Nate Barbettini, OAuth and OpenID Connect in plain English
- https://www.okta.com/identity-101/authentication-vs-authorization/
- https://www.sailpoint.com/identity-library/difference-between-authentication-and-authorization/
- https://jwt.io/
- https://datatracker.ietf.org/doc/html/rfc9126
- https://www.veriff.com/blog/types-of-authentication-methods