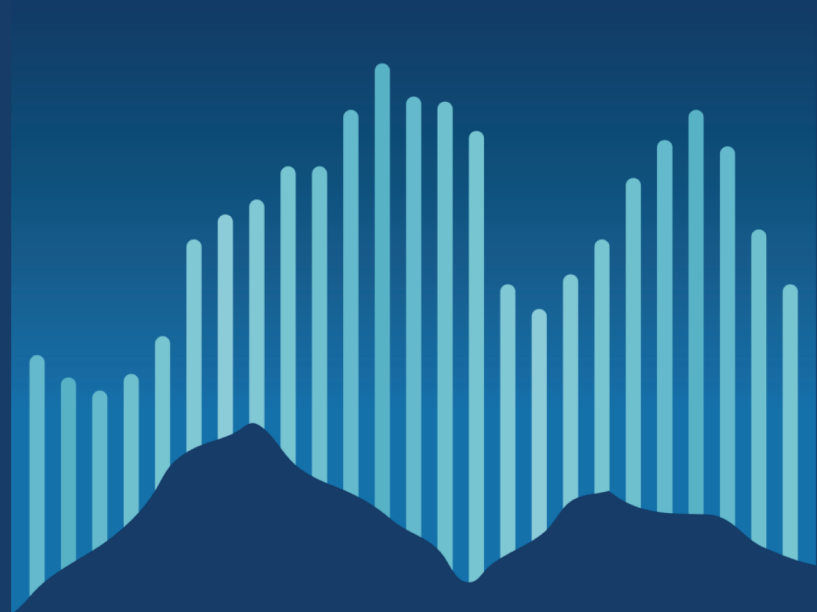


Securing APIs and Microservices with OAuth and OpenID Connect

By Travis Spencer, CEO
[@travisspencer](#), [@curityio](#)



NORDIC **APIS**

- ✓ All API Conferences
- ✓ API Community
- ✓ Active blogosphere

Organizers and founders

2018 Platform Summit

October 22 – 24, 2018 | Stockholm, Sweden

Austin API Summit

TBD 2019 | Austin, Texas

The background of the slide is a dark gray color with a subtle, abstract network pattern. This pattern consists of numerous small, light gray circular nodes connected by thin, light gray lines, creating a complex web-like structure that spans the entire frame. The nodes vary slightly in size, and the lines are thin and light, giving the background a technical or digital feel.

OAuth Basics

Actors, code flow, etc.

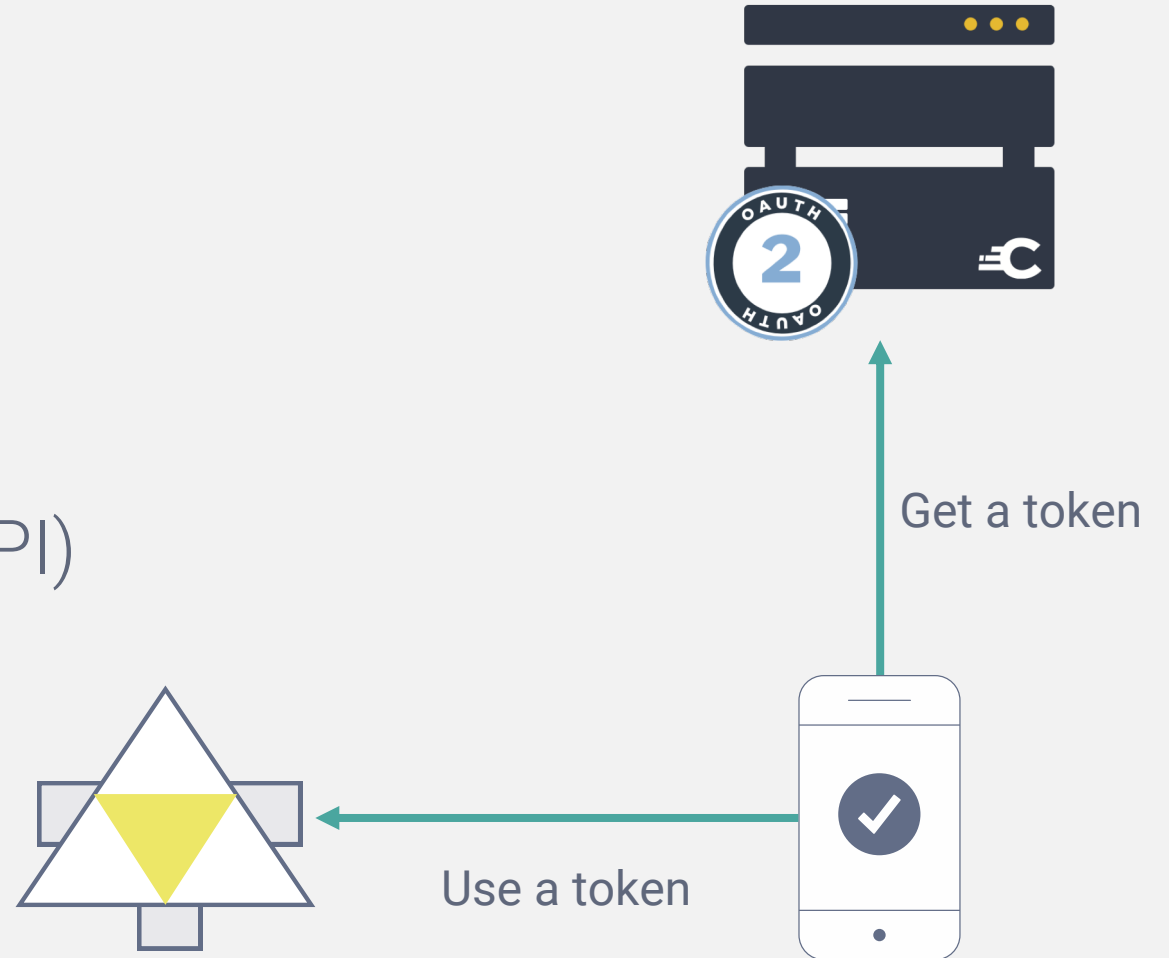
OAuth

- OAuth 2 is a protocol of protocols
 - Used as the base of other specifications
 - OpenID Connect, UMA, HEART, etc.
- Addresses some important requirements
 - Delegated access
 - No password sharing
 - Revocation of access



OAuth Actors

1. Resource Owner (RO)
2. Client
3. Authorization Server (AS)
4. Resource Server (RS) (i.e., API)



Request, Authenticate & Consent



Request Access

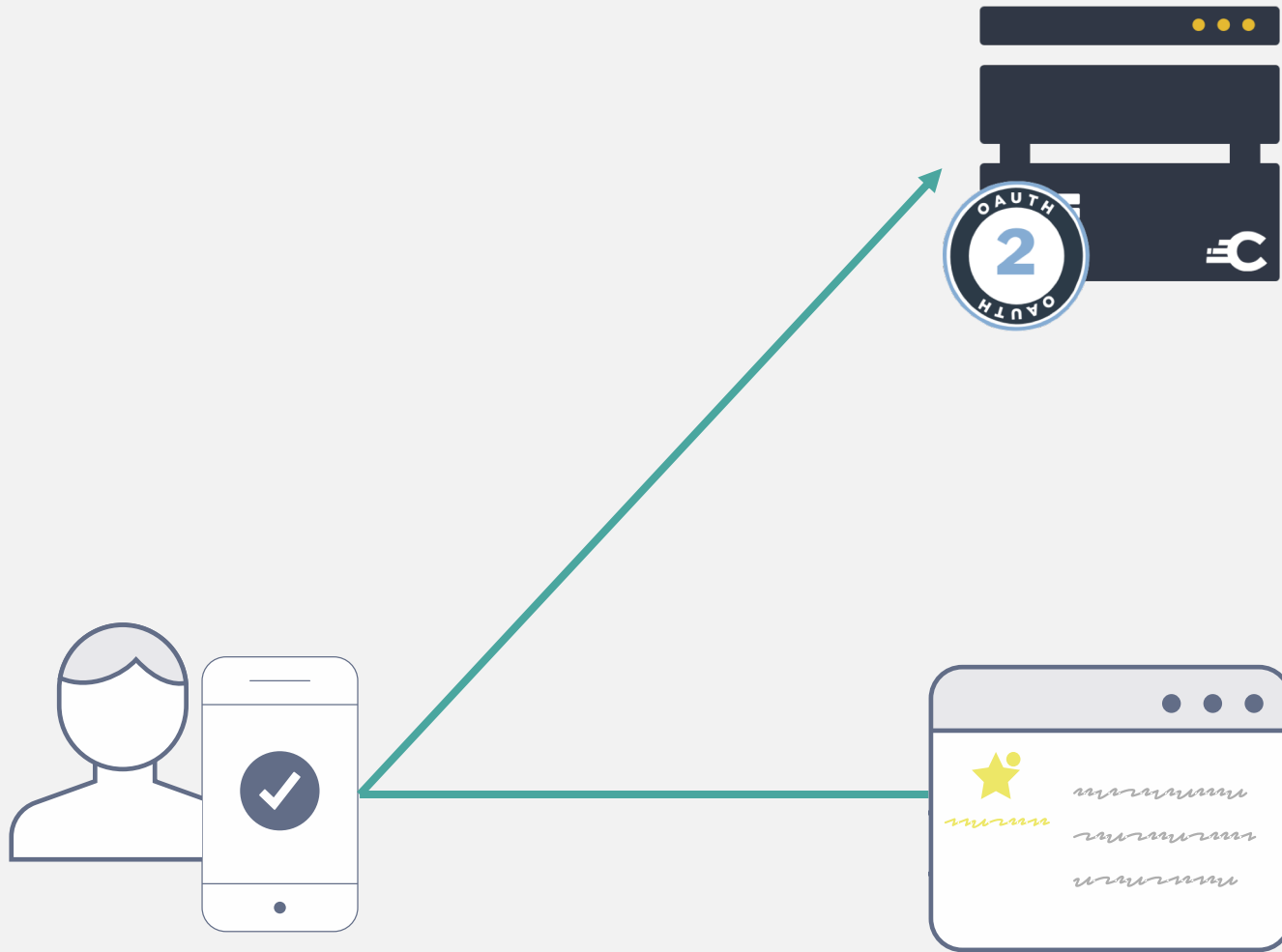


Login



Consent

Code Flow

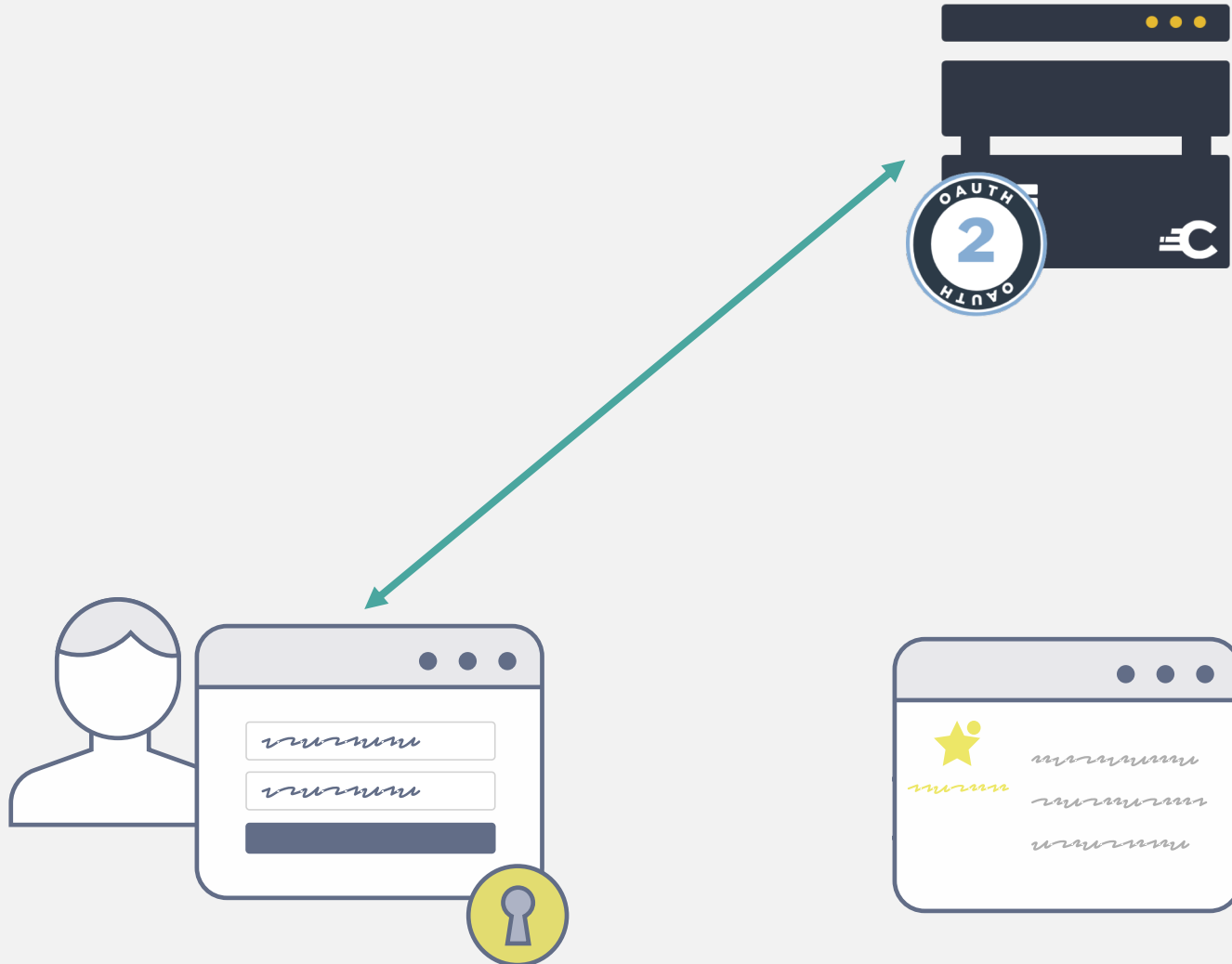


User is redirected to
OAuth server



APIs & microservices

Code Flow

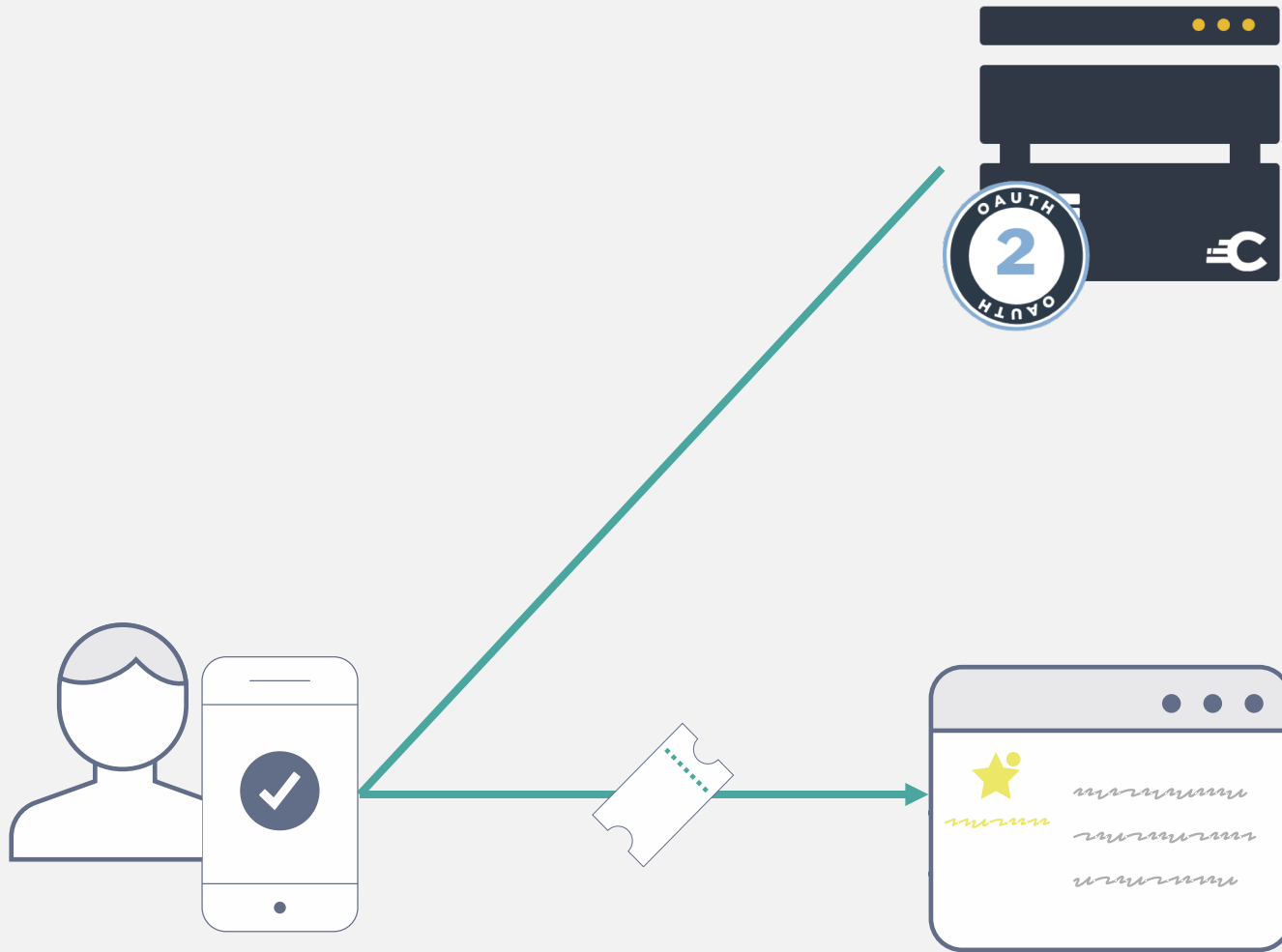


User logs in and delegates access



APIs & microservices

Code Flow

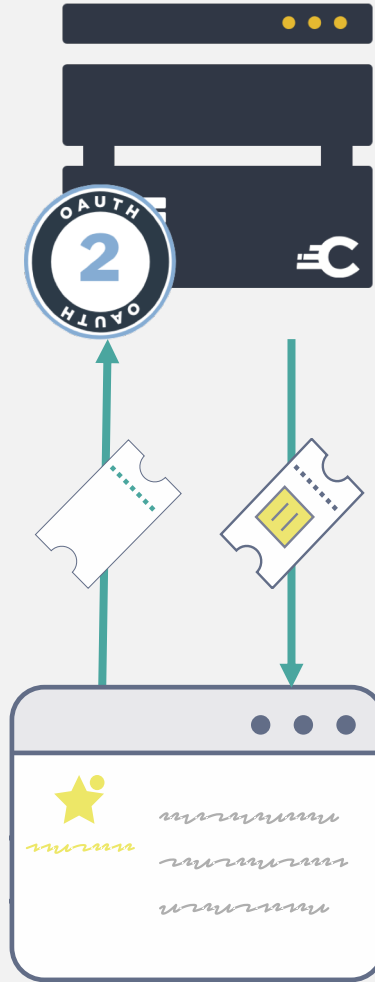
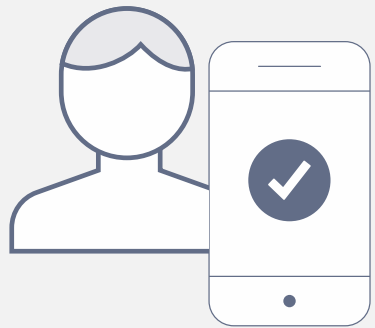


Short-lived access code
is issued to client



APIs & microservices

Code Flow



Code is exchanged for an access token

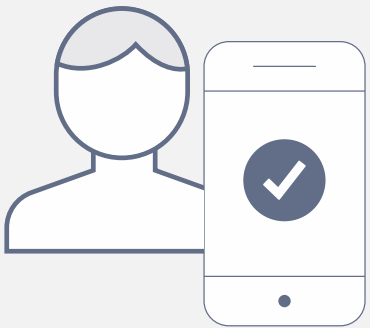


APIs & microservices

Code Flow



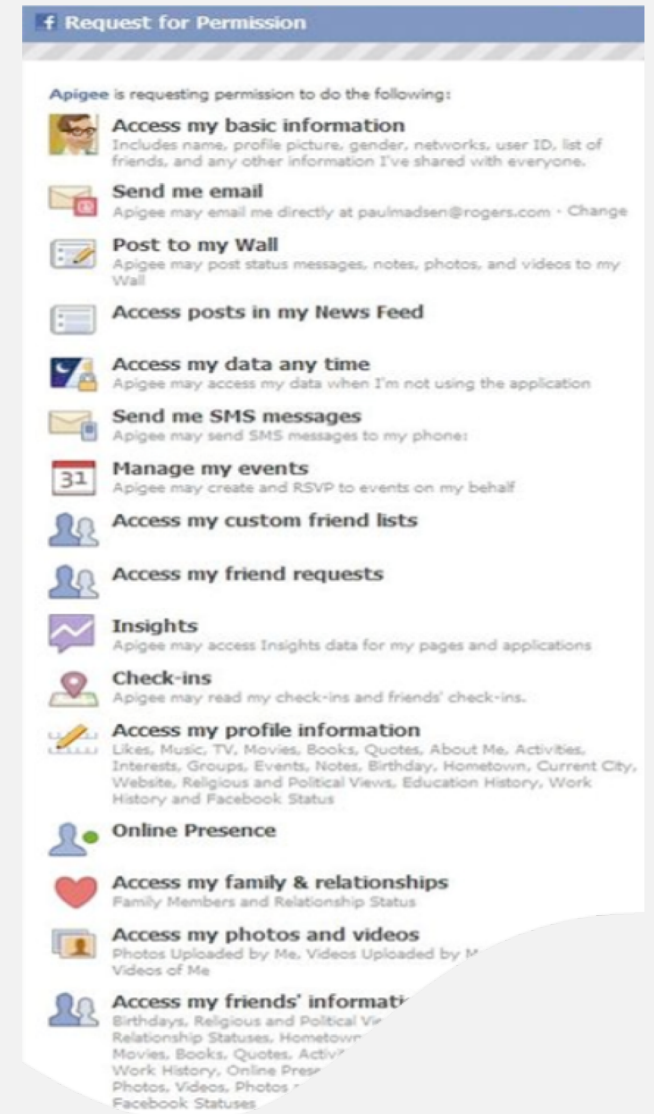
Access token can be used to call APIs



APIs & microservices

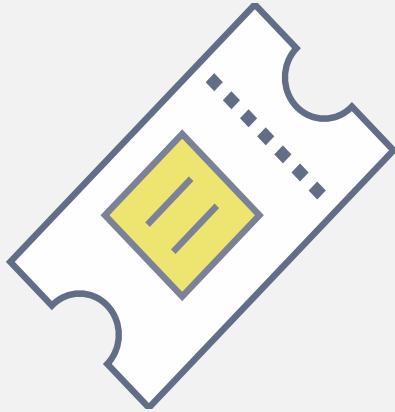
Scopes

- Grouping of claims
- Scope specifies extent of tokens' usefulness
- Claims in scope can be used as permissions
- No standardized scopes (in OAuth); various in OpenID Connect



Kinds of Tokens

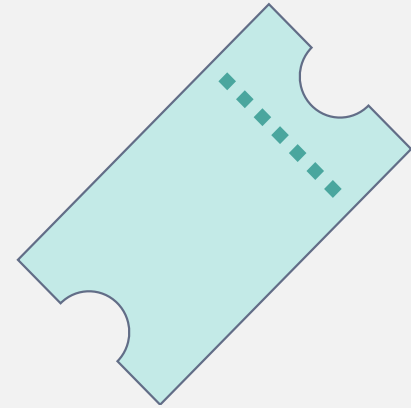
Access Tokens



Like a session

Used to secure API calls

Refresh Tokens



Like a Password

Used to get new access tokens

Profiles of Tokens

Bearer



Bearer tokens are like
cash

Holder of Key



HoK tokens are like
credit cards

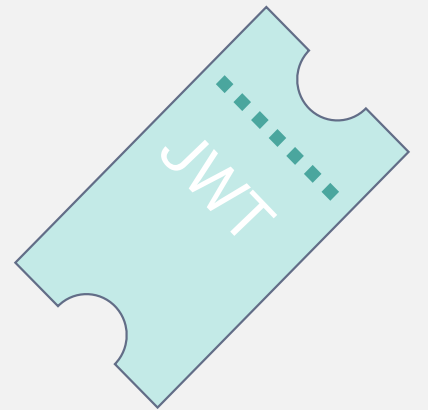
Types of Tokens

- WS-Security & SAML
- Custom
 - Home-grown
 - Oracle Access Manager
 - SiteMinder
- CBOR Web Tokens (CWT)
- JWT



JWT Type Tokens

- Pronounced like the English word “jot”
- Lightweight tokens passed in HTTP headers & query strings
- Encoded as JSON
- Compact
- Encrypted, signed, or neither
- Not the only kind of token allowed by OAuth



Passing Tokens

By Value



User attributes are in
the token

By Reference



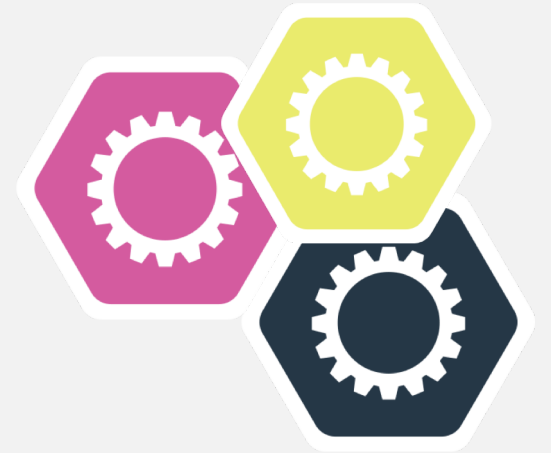
User attributes are
referenced by an
identifier

Improper Usage of OAuth



Not for authentication

Not for federation



Not *really* for authorization

Proper Usage or OAuth



For delegation

The background of the slide is a dark blue-grey color. It features a complex, abstract network of thin, light-grey lines connecting various-sized, semi-transparent grey circles. These circles are scattered across the entire frame, creating a sense of interconnectedness and digital structure. The lines vary in length and orientation, forming a web-like pattern that is more dense in some areas and more sparse in others.

OpenID Connect Overview

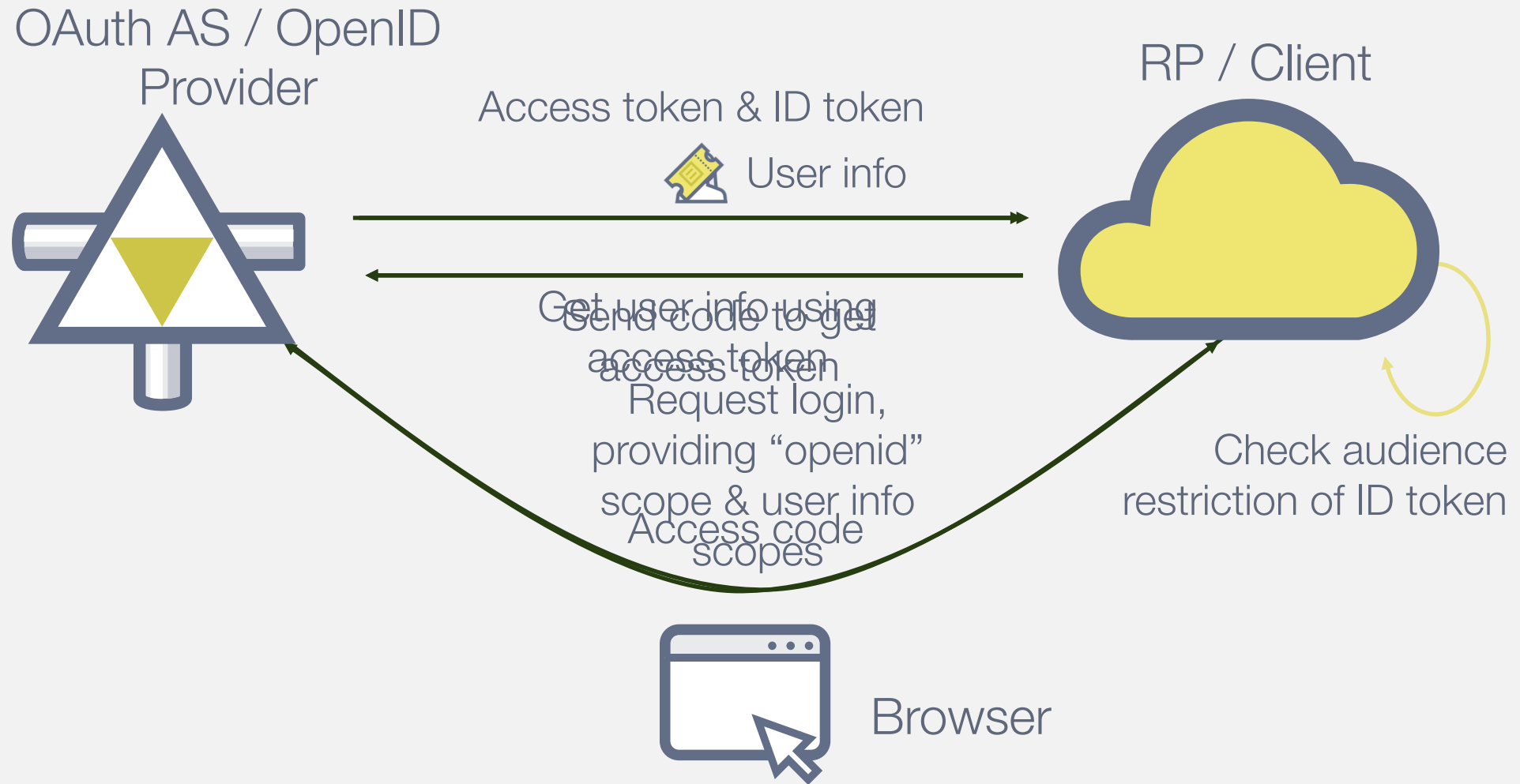
Code flow with ID token, user info, etc.

OpenID Connect

- Based on OAuth 2
- Made for mobile
- Client also receive tokens
- User info endpoint provided for client to get user data
- Additional flows – hybrid
- Claims, request obj, etc.

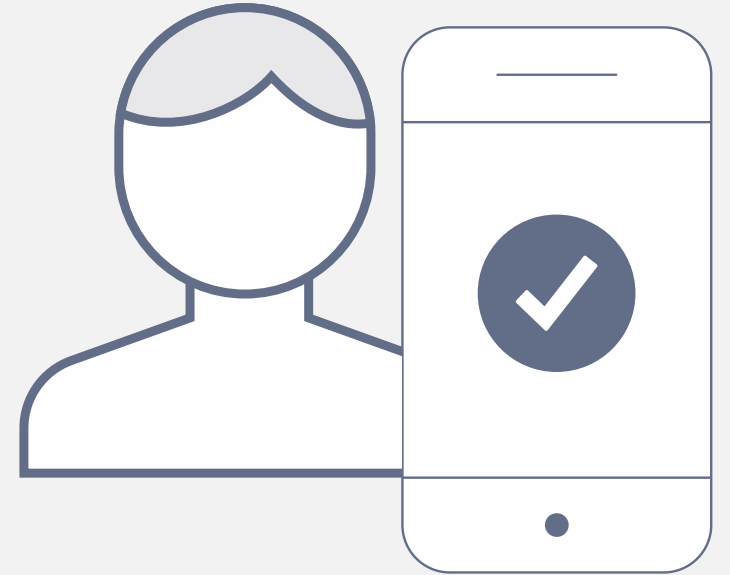


OpenID Connect Examples



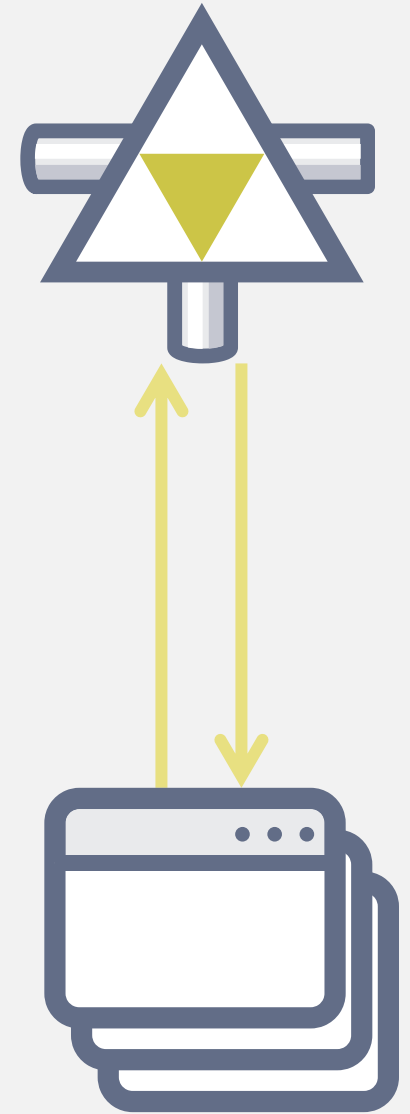
ID Token is for the Client

- Access token is for API
- ID token is for client
- ID token provides client with info about
 - Intended client recipient
 - Username
 - Credential used to login
 - Issuer of token
 - Expiration time

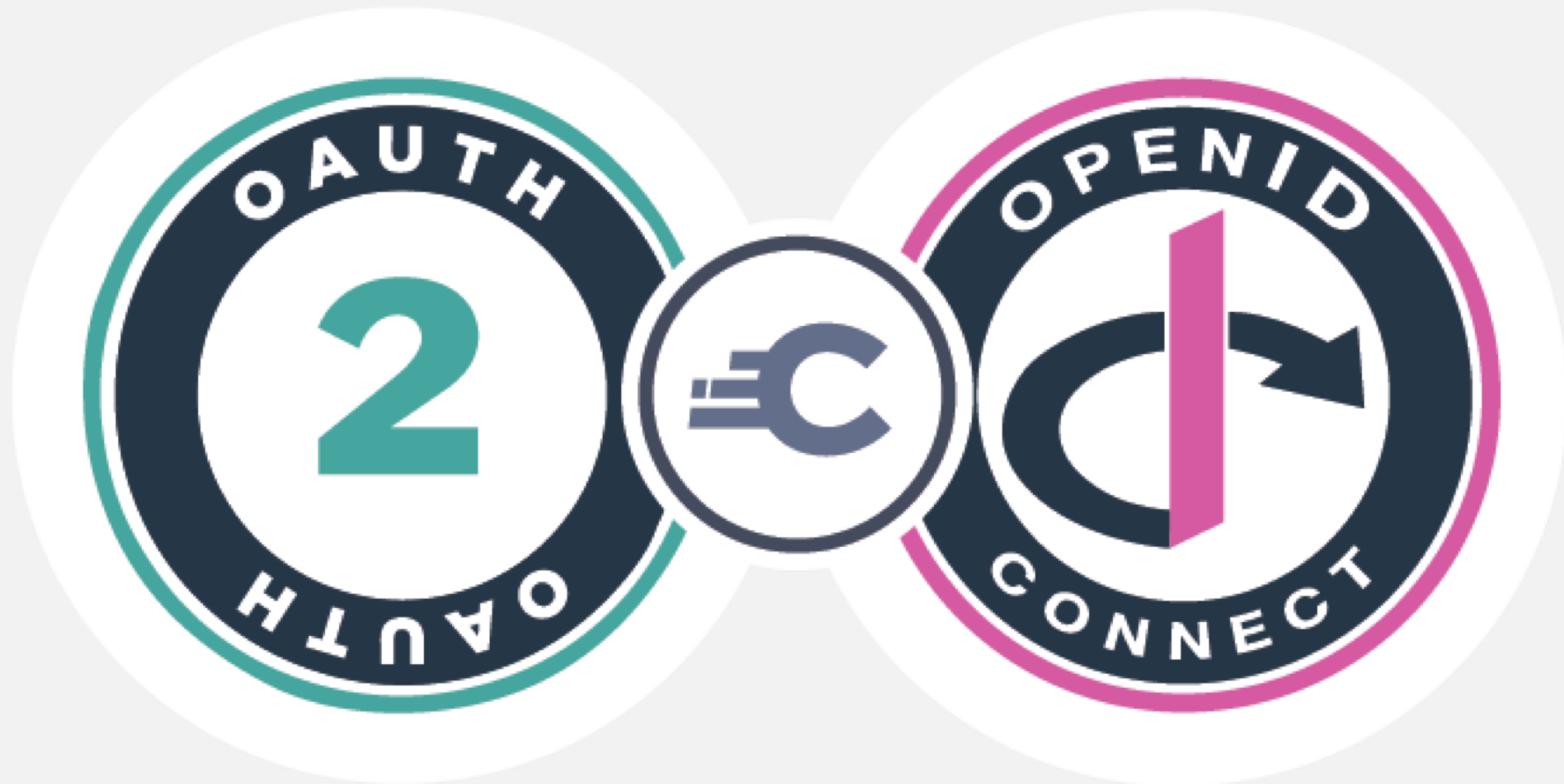


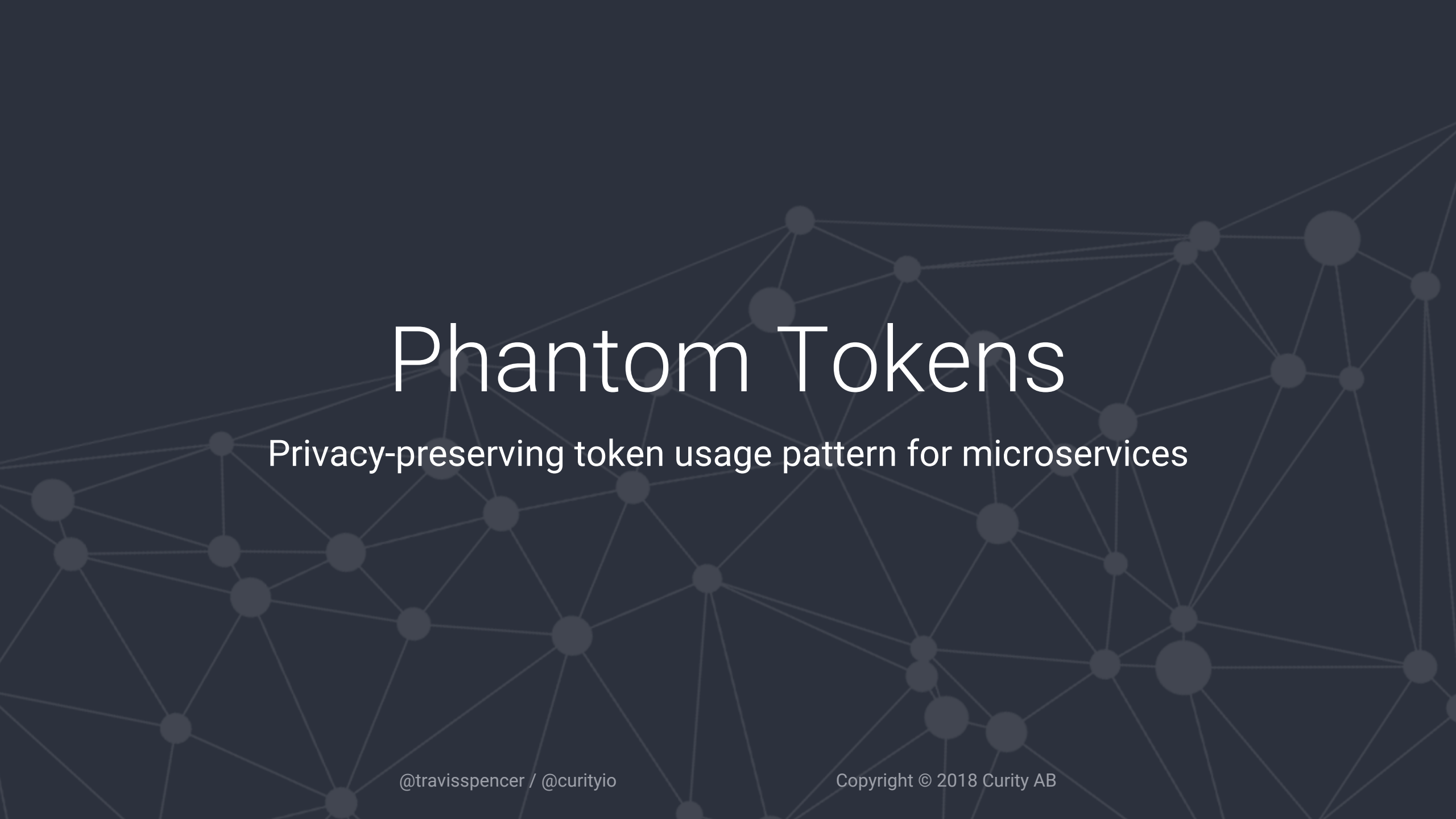
User Info Endpoint

- Token issuance and user discovery endpoint
- Authenticate using access token issued by OpenID Provider
- Output depends on requested and authorized scopes
- sub claim must match sub claim in ID token



Demo

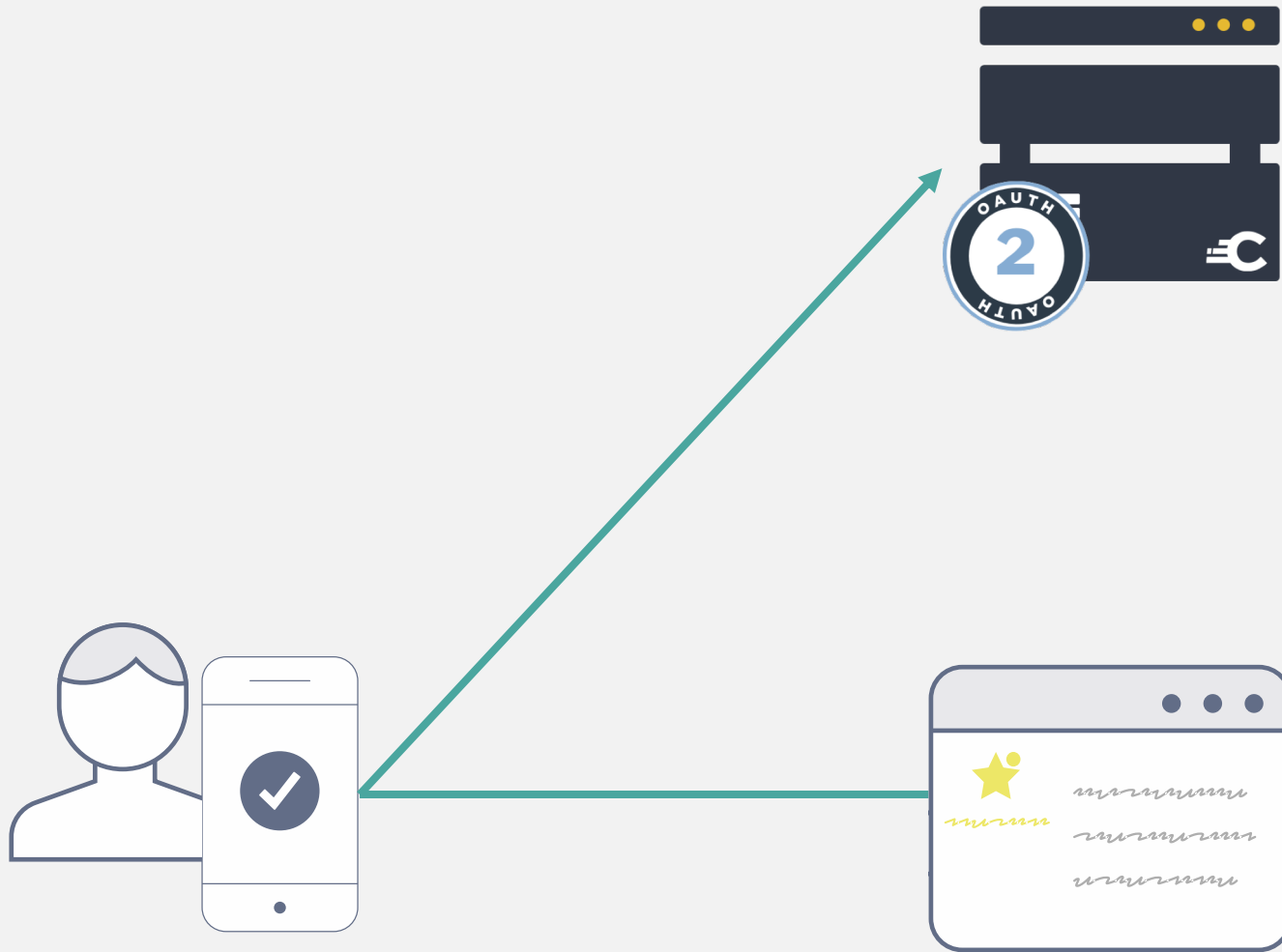


The background of the slide is a dark blue-grey color. It features a complex, abstract network of thin, light-grey lines connecting various-sized circular nodes. The nodes are also in shades of grey, with some being larger than others. The overall pattern resembles a web or a molecular structure, with lines crisscrossing the entire frame.

Phantom Tokens

Privacy-preserving token usage pattern for microservices

Basic Approach

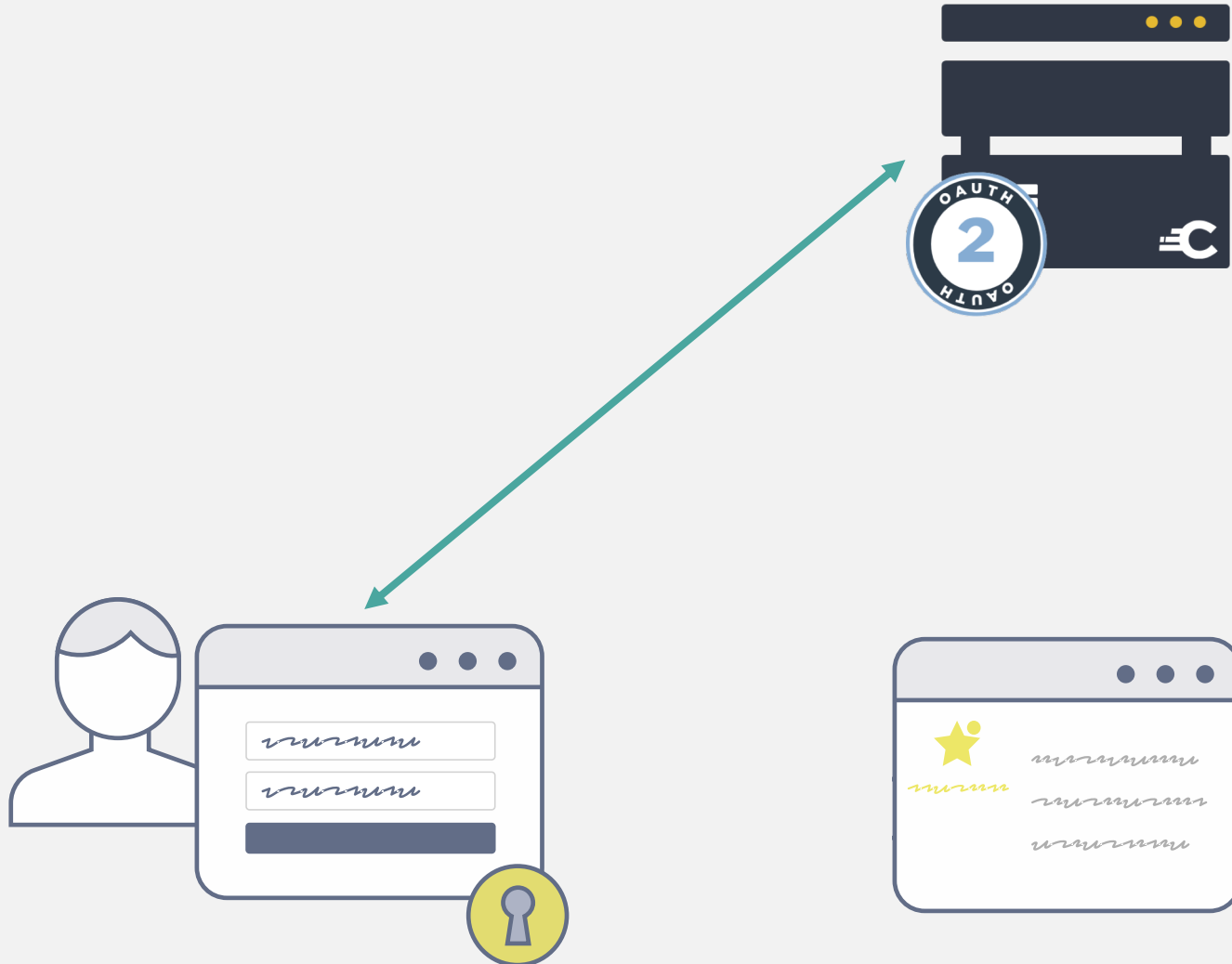


User is redirected to
OAuth server



APIs & microservices

Basic Approach

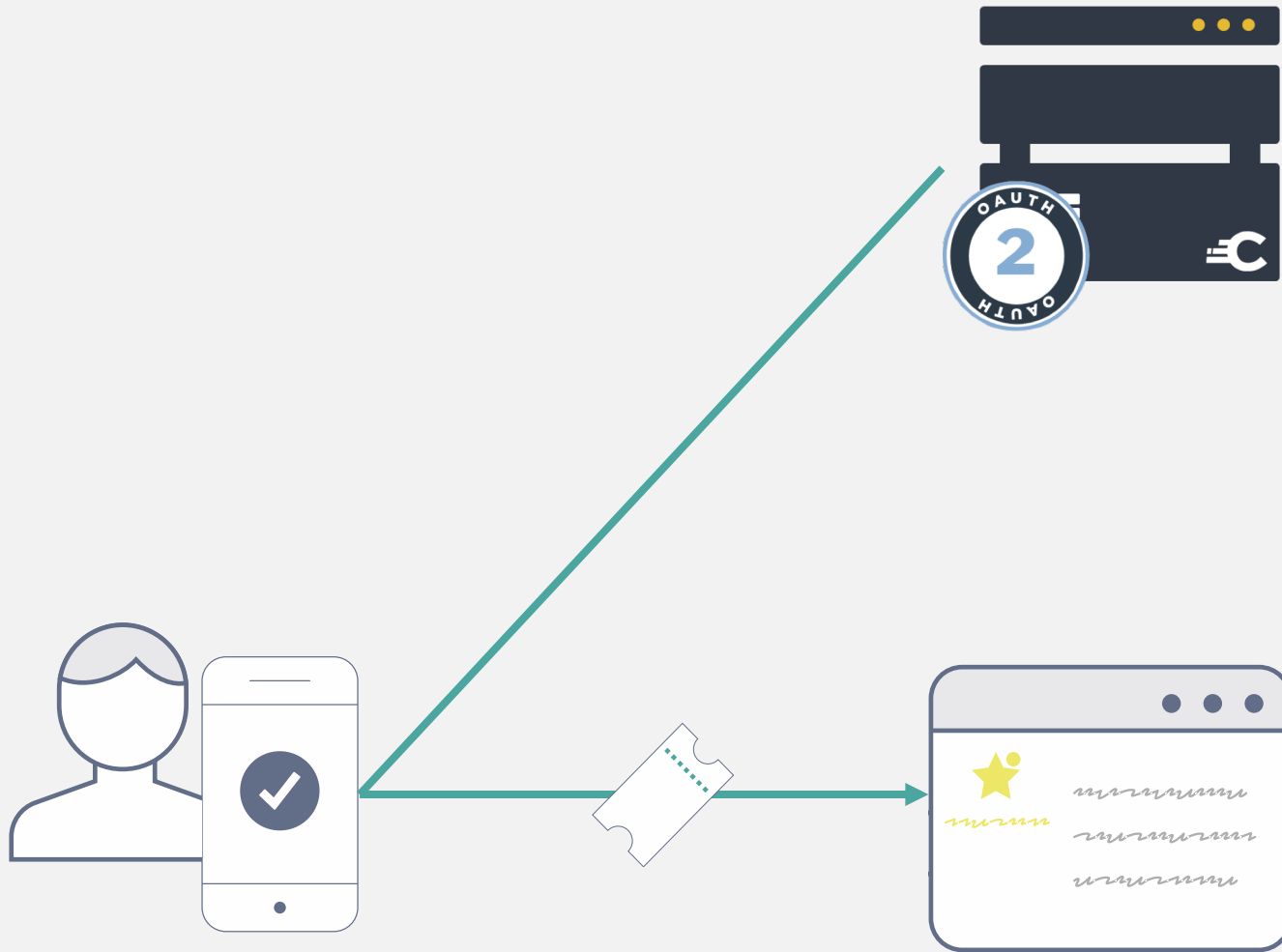


User logs in and delegates access



APIs & microservices

Basic Approach



Access token is issued to client

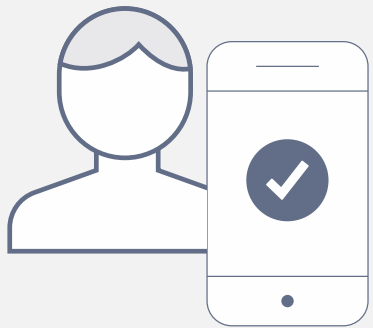


APIs & microservices

Basic Approach



Access token can be used to call APIs

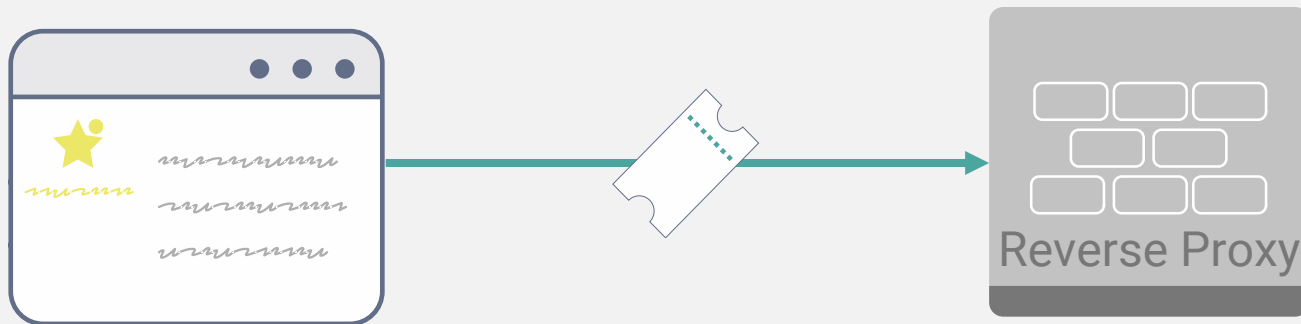


APIs & microservices

Phantom Token Approach

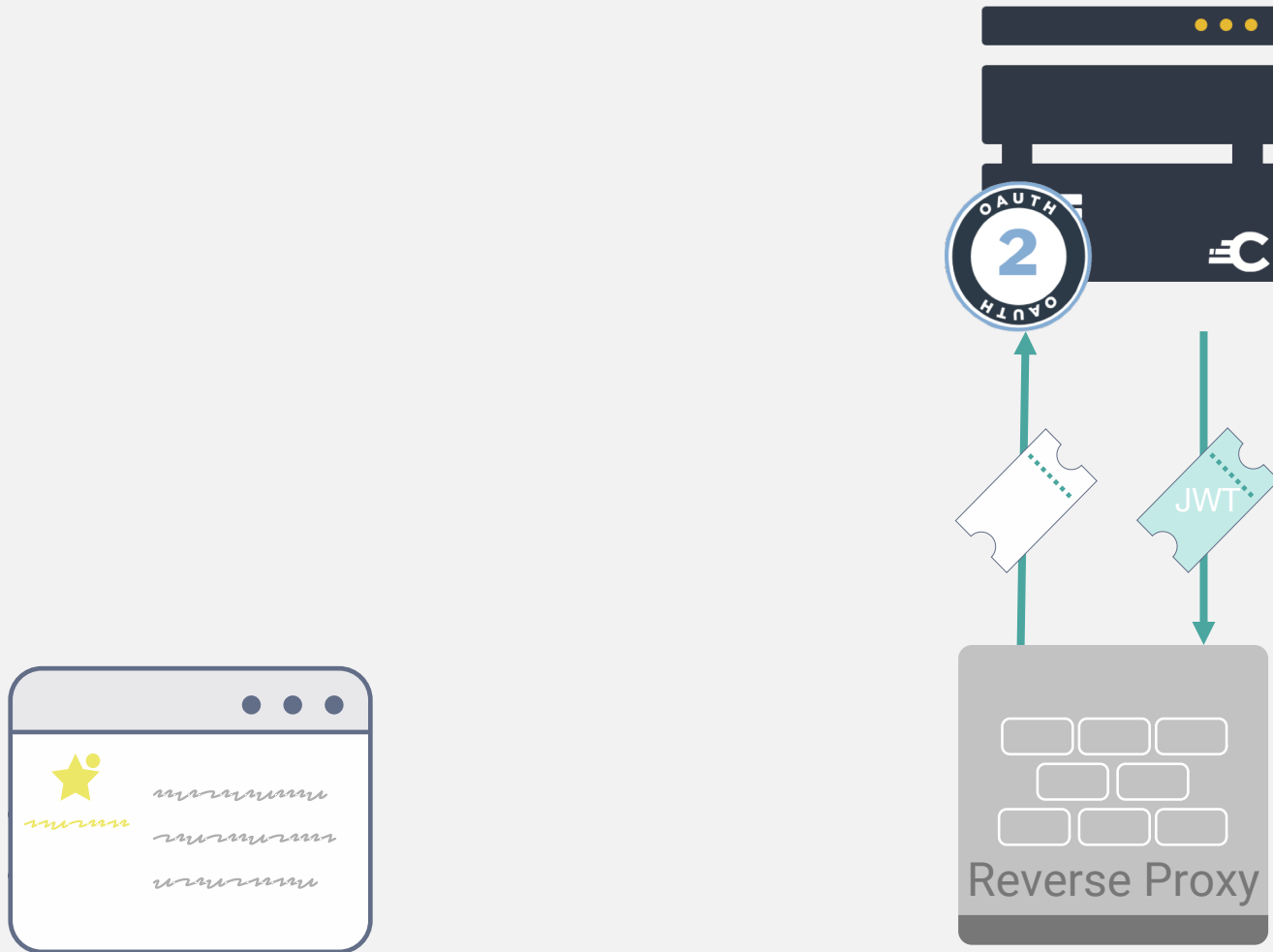


Access token is a by-ref token used externally



APIs & microservices

Phantom Token Approach



External, by-ref token is converted to a by-val internal “phantom token”

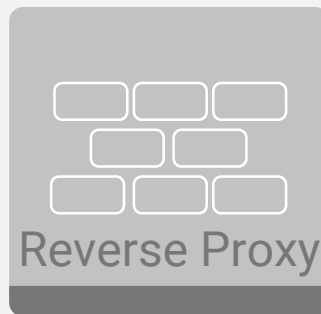
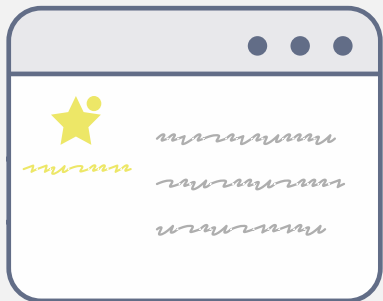


APIs & microservices

Phantom Token Approach



Internal “phantom token”
is used to call actual
APIs



APIs & microservices

Benefits of the Phantom Token Approach

- Smaller regulated space
- Not possible for client to access any PII
 - JWE puts decryption inputs into the attacker's hands
 - JWTs (or other by-val token) only require transport compromise
- Front-end clients can't depend on access token's contents
- Standard compliant
- Vendor neutral

Drawbacks

- Network call to dereference token
 - Cache based on TTL from OAuth server
 - Preemptively populate cache
 - Use a fast reverse proxy
- Doesn't work in multi-tenant scenarios

Alternatives

- By-val JWT that only contains:
 - iss
 - azp
 - exp
 - jti
 - aud

a la HEART

- Must still be introspected

Summary

- Actors – Resource owner and server, client & authorization server
- Flows – Code flow (with and without ID token), etc.
- Token
 - Kinds
 - Ways of passing
 - Profiles
- Appropriate uses – Only delegation
- Use phantom tokens to simplify & secure integration with APIs

More Info

- Come over to our booth and win a Nintendo Switch
- Tutorials on curity.support
- Code examples on github.com/curityio
- Workshops on Thursday
- Download Curity from curity.io





Visit curity.io and stop by our booth