Notes related to Authorization / Authentication

Sunday, June 12, 2016 10:51 AM

Applications which will have Users, when .net was invented

User : Identity
Authorization : Principle } Separation

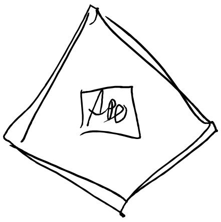
Released 2002

Like AD

User has a Name
Groups User are in Role => Rolebased authorization
Methods are allowed via Group Access

(Intranet + Web)

-> The Applications changed a lot since then
Hops between company, sites, devices
=> PPL think about OAuth
WIF Windows Identity Foundation introduced Claims
white Group / Identity was a yes/no thing Claims allowed you to use key value pairs
Describe with statements, some App may react on the shoe size.
World changed since issuer of Identity and claims are no long the App
AD -> logged into DC
Apps trust the issue



AD -> Name, Role
Google -> First, Last, Mail
ADFS -> Arbitrary AD Properties (Department Location)

Trusted Authority :

Seperate Application from User Login

The Purpose of this service is to sign ppl in :
-> You can better secure and maintain ~~an~~ a service which has only on purpose
What is the return of that service ?

Challenge transfer Identity back to application ~~H~~ -> Protocols (Auth Protocols)
-> AD : Kerberos (Intranet) } XML Signature XML
-> WFederation W*

→ Phones → Open ID Connect ← OAuth first, but:

Based on HTTP
Uses JSON
Krypto easier

API Design
But no Authentication
Just Authorization
⇒ They call it Login
but not <u>really</u>

Open ID is a set of extensions
on top of OAUTH

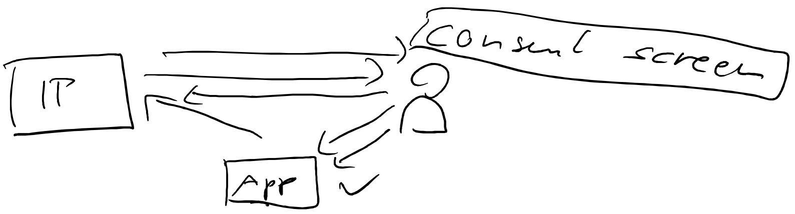
1. Who is the user
2. Access on behalf of the user

IT IS ALL ABOUT
Getting Access Token
for Resource

You get back Identity Token
and even better 1 call
you can also get Identity + Token
in one Roundtrip.

What is a Token?

→ Data Structure
→ Issuer can make sure it is not changed
→ You will get that Token
→ Put in Claims for the user!



Are you okay releasing information to the app.
Last line of defense!
Can you provide that Info to 3rd Parties!

Validate the token as a App

Issuer Name?
Who is this Token for: Client ID?
Signature? Check for manipulation!

What the user can do is up to you. Subject ID
Web App Cookie ✱ !

What the user can do is up to you. Subject ID

↓

Stable Identifier

SID

Web App Cookie ×

Noble Dish

JS Session

→ **APIs** has no cookies and is typically accessed by called apps !

Confusion How do we fill the gap?

⇒ OAuth

Access Token ← Open ID Connect Token, User Identity can throw it away

Identity Token + Access Token

How it works → Issuer (IP)

give me Identity Token → Client

give me Access Token → pass it along to API

HTTP has a Header Property

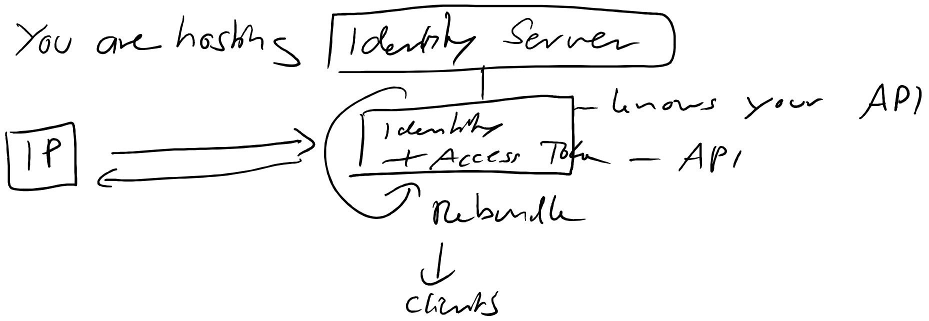
Authorization Header = Transmit Credentials

API put out claims out of the Authorization Header

⇒ Claims Principle

My App + Provider (3rd Party)

Only knows about their API can give you the access Token ⇒ No !



Identity Server – Scope → Name for something you want to call.

Endpoints

/Orders ←

/Management ←

→ Backend ?

What is the difference between

→| User : Human, Carbon based Lifeform

→| Client : The Client which is operated by the User, silicon Lifeform

☑

Identity Server will know by the client !

Which client is allowed to access !

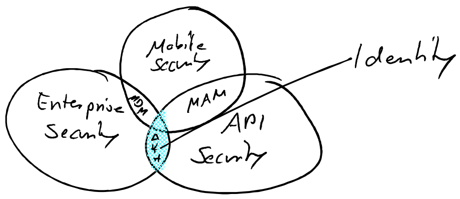
Access Token
-> User
-> Client
-> Scope
} Caller Identity

User and Client can be separated !
-> User can access
-> But particially client not

-> Claims describe the Identity of the user, typically 3, 4 claims !

! Subject ID !

-> You can Revoke Access Tokens



1. How is somebody

-> Federation	(SAML / Open ID Connect)
-> Provisioning	SCIM
-> Identity	JSON Identity Suite
-> Delegated Access	OAUTH
-> Authorization	XACML

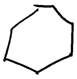
OAuth 2 - is the new protocol of protocols
-> composed in usefull ways
-> Like WS-Trust
- Delegated Access
- No password sharing
- Revocation of access

OAuth Actors — Spec
- client (App)
- Resource Server (API)
- Authorization Server (AS) STS
- Resource Owner (RO)

Scopes
- Like permissions
- Specific extent of tokens usefullness
- Listet on consent UI (if shown)
- Issued tokens may have narrower scope than requested
- No standardized scope

Kind of Tokens

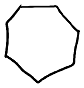
Access Tokens



Like a Session

Invoke API

Refresh Tokens



Like a Password

Used to get new,

Like a Session

Invoke API
(Used to secre API)

Like a Password

Used to get new, Access Tokens

Should never be send to API! They are used to get new Access Tokens!

Passing Tokens

By Value

USER ATTRIBUTES ARE IN THE TOKEN

By Reference

USER ATTRIBUTES ARE REFERENCED BY AN IDENTIFIER

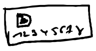
POINTER TO THE DATA

Profile of Tokens

Bearer Tokens

Like Cash

Holder of key



HoK Tokens are like Credit Cards

Have to check when it is presented.

Types of Tokens

- WS-Security
- SAML
- JWT
- Custom
  - Home-grown
  - Oracle Access Manage
  - Site Minder
- ETC

JSON Identity Protocol Suite — Being defined in IETF

Suite of Json-based Identity protocols

- Tokens (JWT) – Lightweight (URL) ~
- keys (JWK)
- Algorithms (JWA)
- Encryption (JWE)
- Signatures (JWS)

Less Expressive

JSON instead XML

More Compact