Notes related to Authorization / Authentication

Sunday, June 12, 2016 10:51 AM

Applications which will have Use

User : Identity
Authorization : Principle } Separation

Like AD
User has a Name
Groups User are in Role => Role

(Internet + Web)

The Applications changed a lot sin
-> Hops between company, sites, devices
=> PPL think about OAuth
WIF Windows Identity Foundation introduced
white Group/Identity was a yes/no thing
use key value pairs
Describe with statements, some App
world changed since issuer of identity
AD -> logged into DC
Apps trust the issue

AP -> Name, Role
Google -> First, Last, Mail
ADFS -> Arbitrary AD Properties (T

Trusted Authority !

Separate Application from User Logi

The Purpose ~~of this~~ service is to sig

→ You can better secure and mai
only on purpose

What is the return of that s

Challenge transfer Identity back to applicati

→ AD : Kerberos (Intranet)

→ WFederation

→ Phones → OpenID Connect ←
Based on HTTP
uses JSON
keypo easier

OpenID is a set of extensions
on top of OAUTH

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

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 c
→ You will get that Token
→ Put in Claims for the user ☺

JS

Disk
Session

→ APIs has no cookies and is typically
Confusion How do we fill the gap?

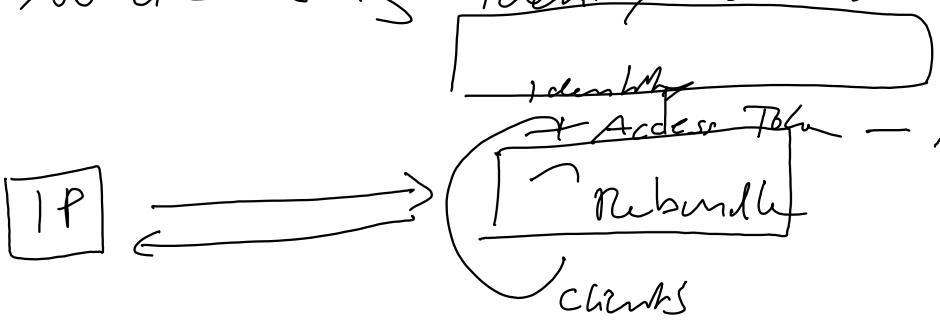
=> OAuth

Access Token ← open ID Connect
Token, Use Identity
Identity Token
+ Access Token

How it works => Issuer (IP)
give me Identity Token → Client
give me Access Token → Pass it along:
HTTP has a He
Authorization He
API put out
Authorization
=> Claims Pr

My App + Provider (3rd Party)
Only knows about their API
can give you the access Token =

You are hosting Identity Server



Identity Server — Scope > Name for so
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

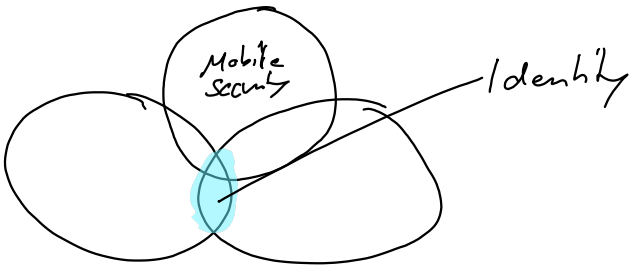
User and Client can be seperated!

→ User can access
→ But partically client act

→ 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

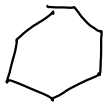
OAuth 2 - is new protocol of protocols
→ composed in usefull ways
→ like WS-Trust
- Delegated Access
- No password sharing
- Revocation of access

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

Scopes
- like permissions
- Specific extent of tokens usefullness
- Listed 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
(Used to secure API)