

# MSc - Cybersecurity

## CMT310: Developing Secure Systems and Applications

**Authentication, OAuth, Open ID**

Dr Neetesh Saxena  
[saxenan4@cardiff.ac.uk](mailto:saxenan4@cardiff.ac.uk)

# Outline

- Authentication: Types and Techniques
- OAuth
- SAML-based Authentication
- OpenID and OpenID Connect

# Definition - Authentication

- Authentication is the
  - **process of validating the identity** of someone or something.
- Generally authentication requires the
  - **presentation of credentials or items of value** to really prove the claim of who you are.
- The items of value or credential are based on several **unique factors** that show
  - *something you know, something you have, or something you are.*

# Definition - Authentication

- ***Something you know:*** This may be something you mentally possess. This could be a *password*, a *secret word* known by the user and the authenticator.
- ***Something you have:*** This may be any form of issued or acquired self identification such as:
  - SecurID
  - CryptoCard
  - Activcard
  - SafeWord
  - and many other forms of cards and tags.
- ***Something you are:*** This being a naturally acquired physical characteristic such as *voice*, *fingerprint*, *iris pattern* and *other biometrics*.
- In addition to the top three factors, another factor, though indirect, also plays a part in authentication.
  - ***Somewhere you are:*** This usually is based on either physical or logical location of the user. The use, for example, may be on a terminal that can be used to access certain resources.

# Authentication Methods

- Password, public-key, anonymous and certificate-based authentication.
  - **Password Authentication:**
    - includes reusable passwords, one-time passwords, challenge response passwords, and combined approach passwords.
  - **Public Key Authentication:**
    - generate a pair of keys
      - usually between 1024 and 2048 bits in length.

**Key escrow** - the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys. These third parties may include businesses, who may want access to employees' private communications, or governments, who may wish to be able to view the contents of encrypted communications.

# Authentication Methods Cont.

- **Anonymous Authentication:**
  - do not have membership to the system they want access to.
  - access the system via a special “anonymous” account.
- **Digital Signatures-based Authentication:**
  - does not require passwords and user names.
  - consists of an electronic signature to verify user identity
    - Digital Signature Algorithm (DSA),
    - Elliptic Curve Digital Signature and Algorithm (ECDSA),
    - account authority digital signature.

# Authentication Threats

- Cellular network
  - GSM
- Sessional authentication
  - Citi bank – login, change address bar id number.
- Apple iCloud – interfaces (file, photos,... find my iphone)
  - Find My iPhone - for locating lost or stolen iPhones and iPod.
  - You can even remotely delete all the data from your phone.
  - Make as many attempts as you want for Apple ID and password...
- Webpage scripting: client-side checks

# Authentication Threats Cont.

- Reset by answering security questions
- Reset by gmail – (single signon)
  - Get an email on alternative email ID (solution)
- Recover and Reset
  - Recover for foo@me.com
  - Allow you to reset password – address and last 4 digit of CC#
- Amazon (old system)
  - Add CC#
  - Make payment with your login and other's card



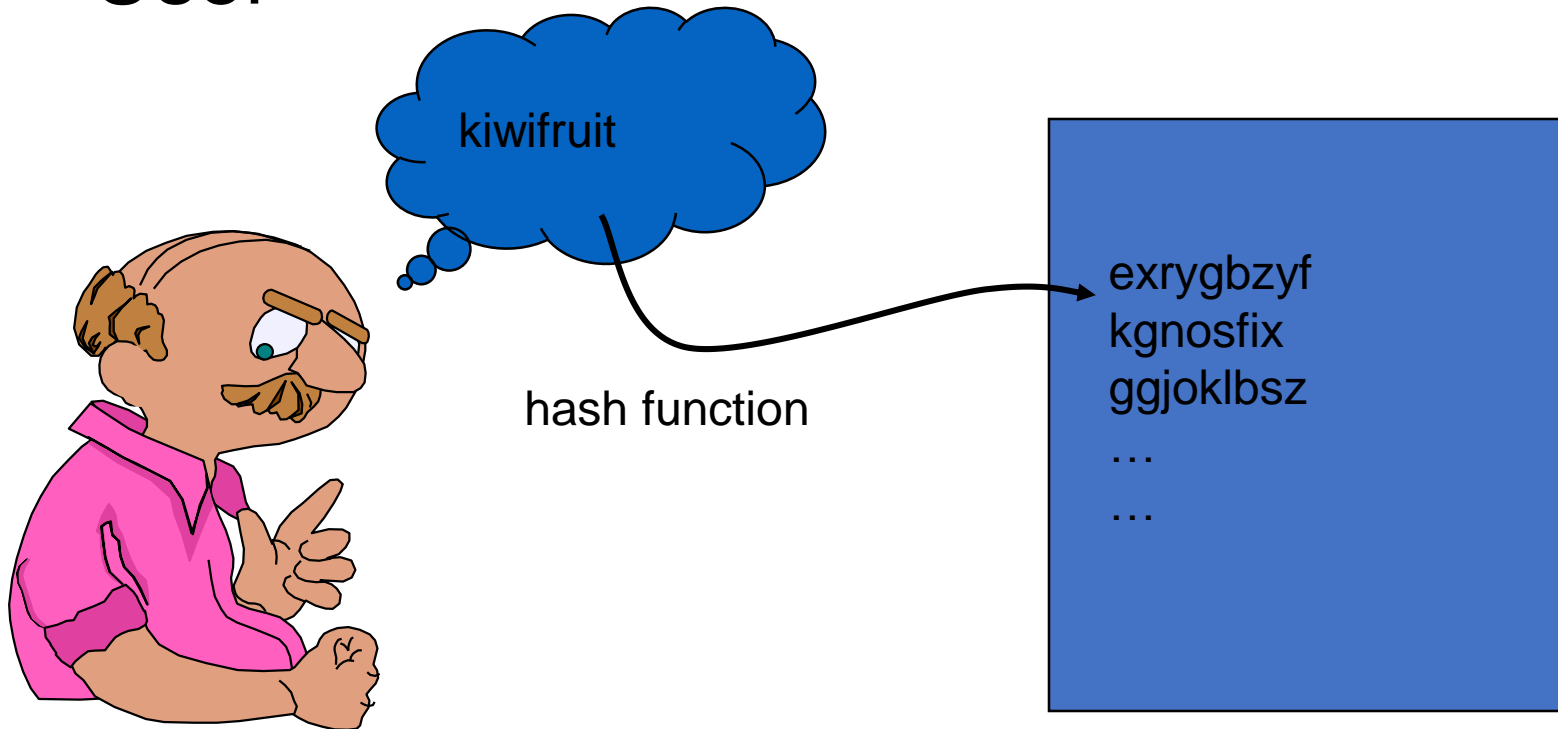
# Authentication Methods

- Text-based passwords
- Graphical passwords
- Hardware tokens
- Public key crypto protocols (already covered)
- Biometrics
  - Biological
  - Behavioral

# Password File

User

Password file



# Passwords: Types of Password Cracking

- Dictionary Attack
  - Quick technique that tries every word in a specific dictionary
- Hybrid Attack
  - Adds numbers or symbols to the end of a word
- Brute Force Attack
  - Tries all combinations of letters, numbers & symbols
- Popular programs for Windows password cracking
  - LophCrack
  - Cain & Abel
  - John the Ripper
  - SamInside

# Passwords: How Can Passwords Be Stored?

 Filing System  
Clear text




 Dedicated Authentication Server  
Clear text



 Encrypted

 Password + Encryption = bf4ee8HjaQkbw

Hashed


 Password + Hash function =  
aad3b435b51404eeaad3b435b51404ee

Salted Hash

(Username + Salt + Password) + Hash function =  
e3ed2cb1f5e0162199be16b12419c012

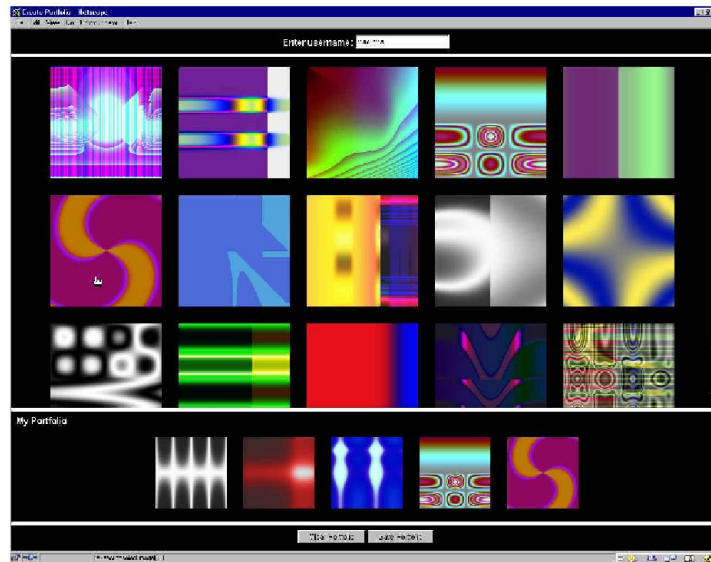
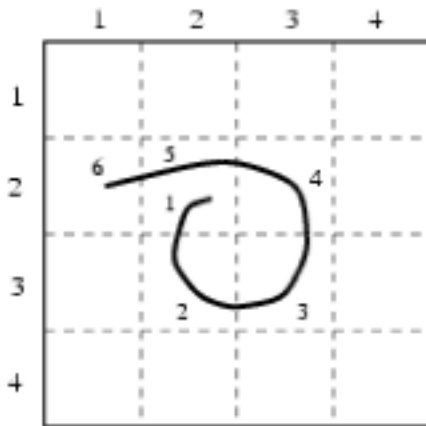
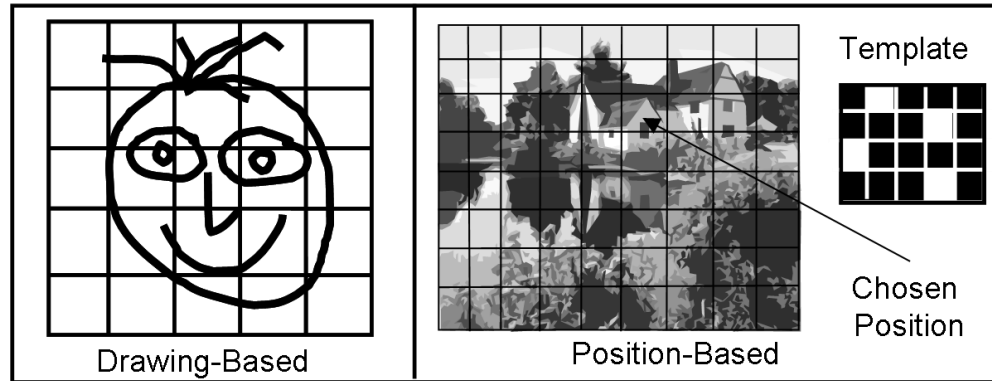
# Improving Security

- Password complexity
  - Case-sensitivity
  - Use of special characters, numbers, and both upper and lower-case letters
  - Minimum length requirements
- Security questions
  - Ask personal questions which need to be verified
  - Some questions are very easy to discover answers
- Virtual keyboard
  - Person clicks on-screen keyboard to enter password (prevents keylogging)
- Single sign-on
  - User only has to remember one password at a time and yet can access all/most of their resources
  - AKA Enterprise Reduced Sign-On
- Centralized password storage management
  - Online sites accessible through one password which contain all other passwords



Single point  
of failure, but  
easier to  
remember

# Graphical Passwords



# Pass Points

- Use “a sequence of clicks” as a shared secret
- There are hot spots



# Personal Token Authentication

- Personal Tokens are hardware devices that generate unique strings that are usually used in conjunction with passwords for authentication.
- A variety of different physical forms of tokens exist
  - e.g. hand-held devices, Smart Cards, USB tokens.
- Different types of tokens exist:
  - Storage Token:
    - A secret value that is stored on a token and is available after the token has been unlocked using a PIN.
  - Synchronous One-time Password Generator:
    - Generate a new password periodically (e.g. each minute) based on time and a secret code stored in the token.
  - Challenge-response:
    - Token computes a number based on a challenge value sent by the server.



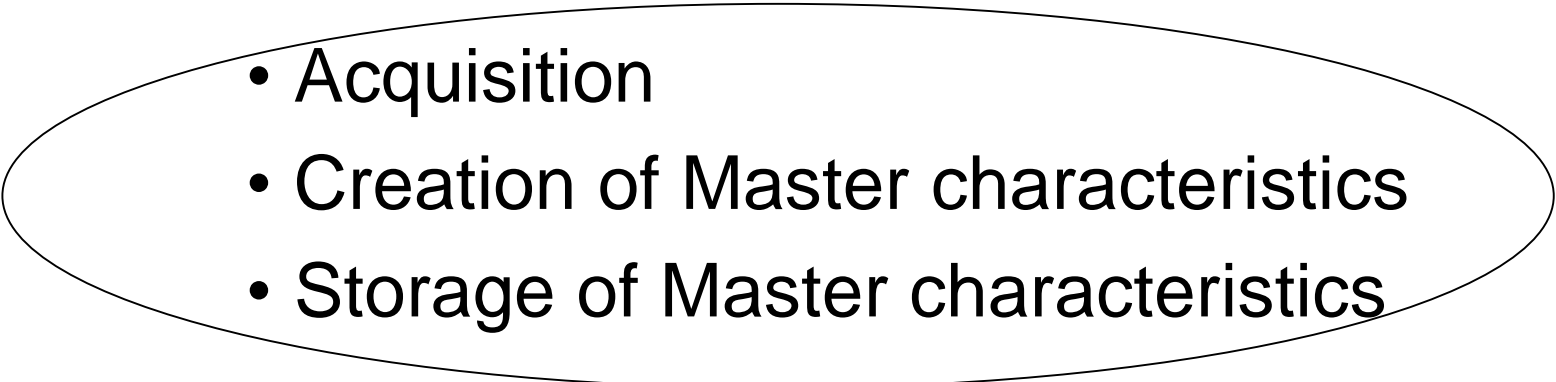


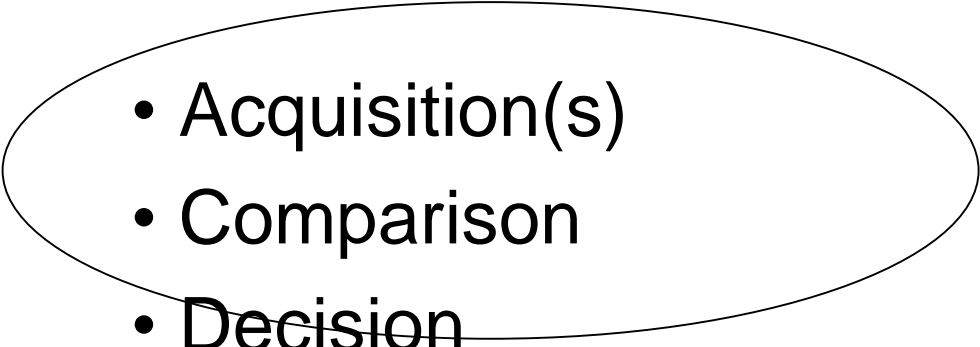
# Biometric Authentication

- Biological:
  - Fingerprint, Iris, Retina, Face, and Hand Recognition.
- Behavioral:
  - Handwriting, Gait, Voice, Signature, Typing pattern, Mouse Gesture Recognition.



# Biometric Authentication Process

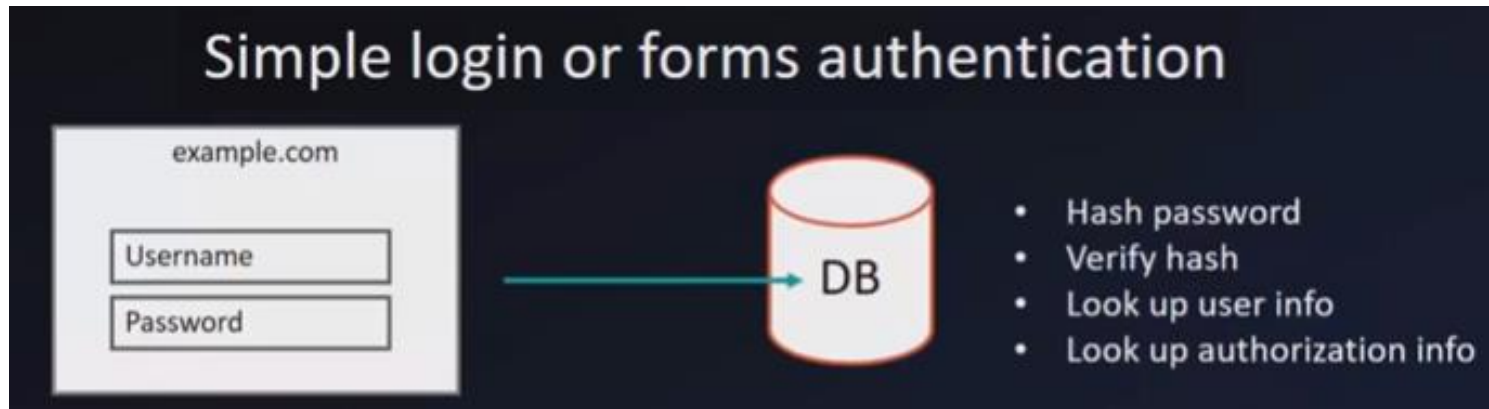
- 
- Acquisition
  - Creation of Master characteristics
  - Storage of Master characteristics

- 
- Acquisition(s)
  - Comparison
  - Decision

# Think

- Password vs. biometric?
- Why banks block the user login after few unsuccessful attempts?

# Authentication Problem



## Identity use cases (pre-2010)

- Simple login (forms and cookies)
- Single sign-on across sites (SAML)
- Mobile app login (???)
- Delegated authorization (???)

## Downsides

- Security
- Maintenance

OAuth 2.0 and OpenID Connect are becoming the industry best practices for solving these problems.

# OAuth



John is a developer who has been told he needs to use OAuth 2.0 to authorize users with an external server.

## OAuth 2.0 Roles



User



Application



API

Cont.

# OAuth 2.0 Roles



User



Application



API



*"I get it now. When I log into Spotify with my Facebook account, Spotify grabs my username and password from Facebook. So easy!"*

Passwords are never passed from server to server in an OAuth 2.0 framework.

Cont.

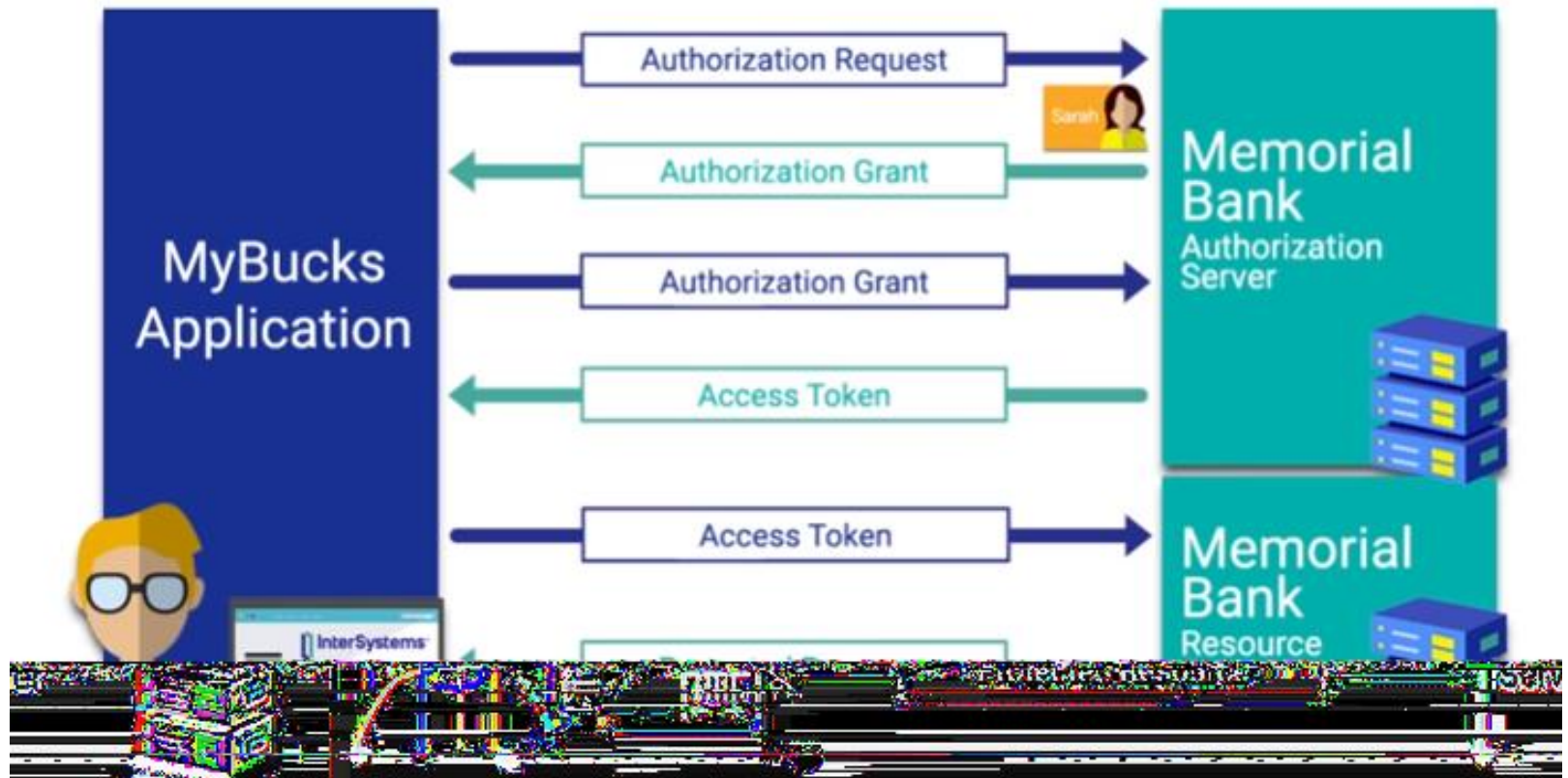
# Workflow of OAuth 2.0





Cont.

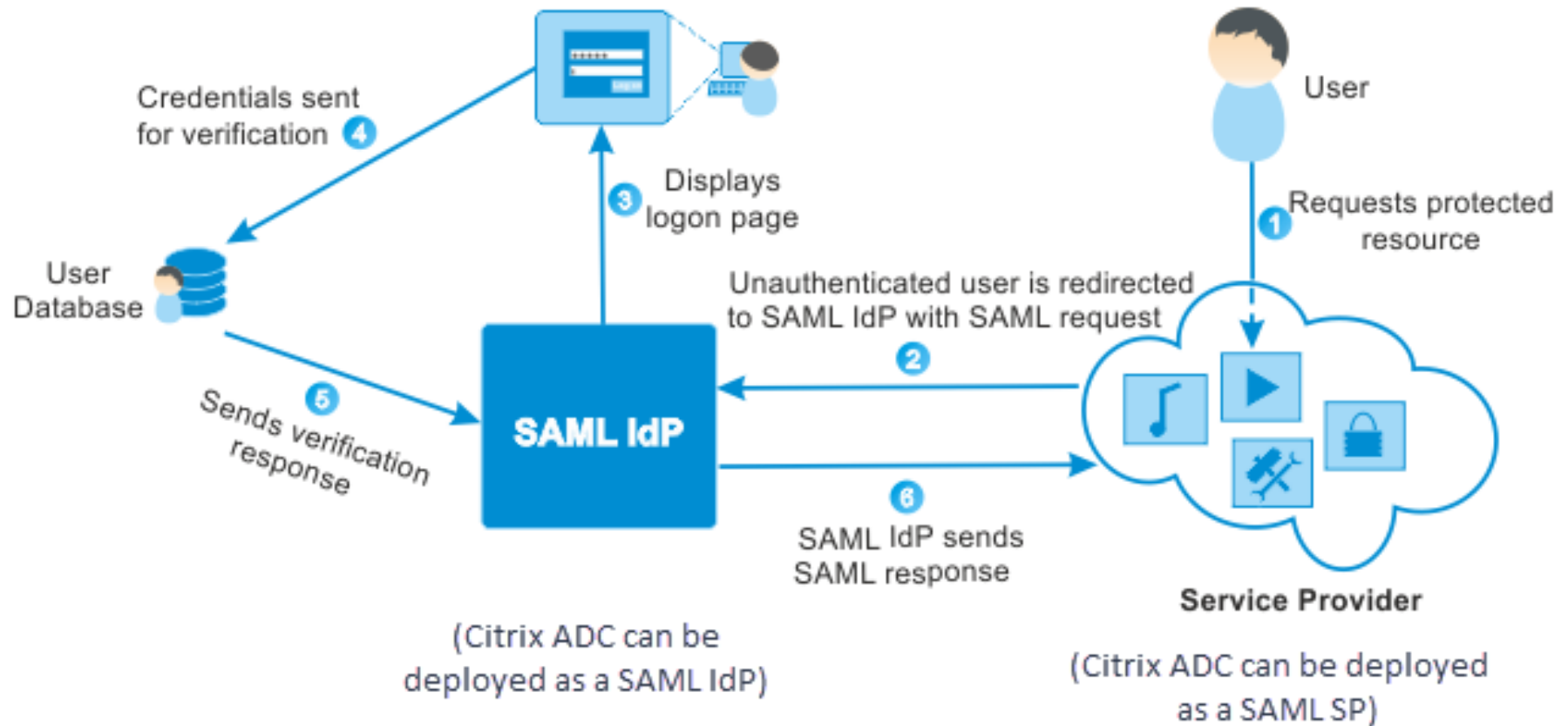
# Workflow of OAuth 2.0



Token access happens at the back channel – cannot steal/eavesdrop token.  
Cannot trust browser/user interaction front channel.



# Security Assertion Markup Language (SAML)-based Authentication



**SAML:** open standard for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider. SAML is an XML-based markup language for security assertions (statements that service providers use to make access-control decisions).

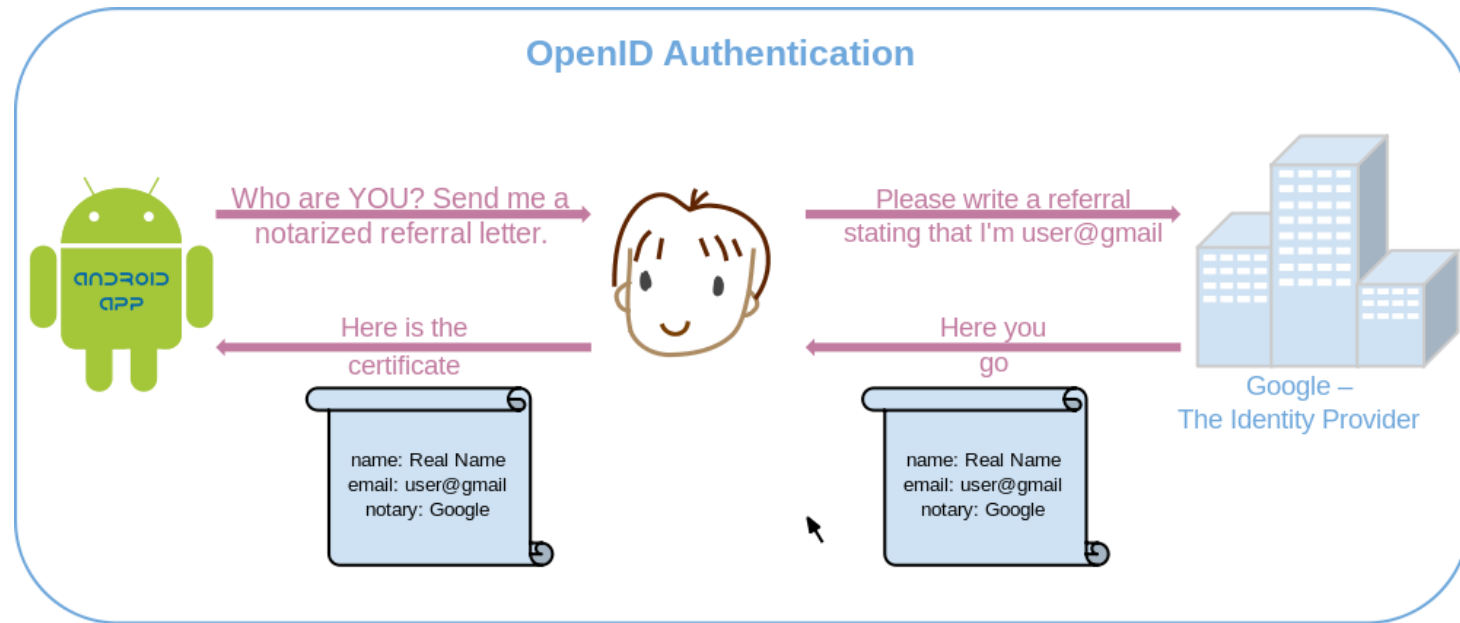
**Example:** webpage single sign-on (SSO)

**Citrix ADC:** application delivery controller (ADC) that accelerates application performance, enhances application availability with advanced load balancing, secures mission-critical apps from attacks and lowers server expenses by offloading computationally intensive tasks.

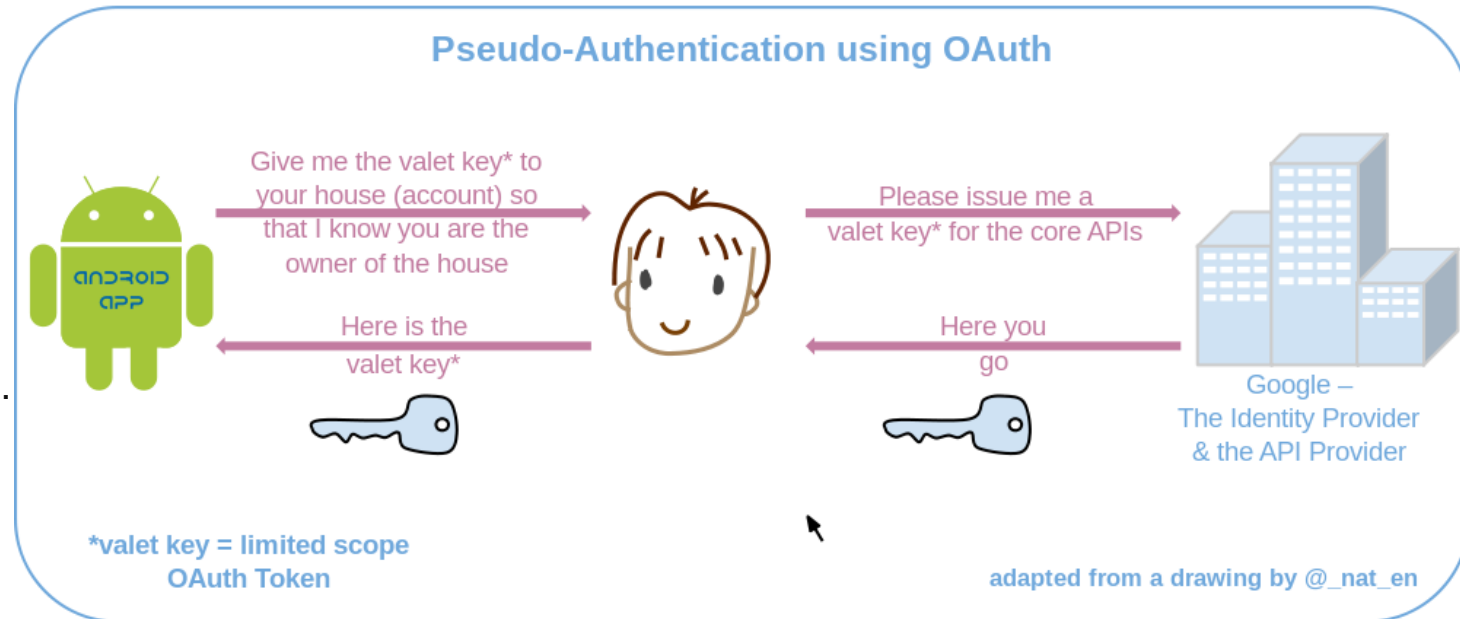
OpenID is about authentication (ie. proving who you are).

OAuth is about authorisation (ie. to grant access to functionality/data/etc.. without having to deal with the original authentication).

There are two popular industry standards for Federated Authentication. SAML (or Security Assertion Markup Language) flow, and OpenId Connect.



VS.



# Examples

- **Scenario for OpenID**

- User wants to access his account on example.com
- example.com (the “Relying Party” in OpenID lingo) asks the user for his OpenID
- User enters his OpenID
- example.com redirects the user to his OpenID provider
- User authenticates himself to the OpenID provider
- OpenID provider redirects the user back to example.com
- example.com allows the user to access his account

- **Scenario for OAuth**

- User is on example.com and wants to import his contacts from mycontacts.com
- example.com (the “Consumer” in OAuth lingo) redirects the user to mycontacts.com (the “Service Provider”)
- User authenticates himself to mycontacts.com (which can happen by using OpenID)
- mycontacts.com asks the user whether he wants to authorize example.com to access his contacts
- User makes his choice
- mycontacts.com redirects the user back to example.com
- example.com retrieves the contacts from mycontacts.com
- example.com informs the user that the import was successful

# OAuth vs. OpenID vs. OpenID Connect

- **OAuth** - only provides authorization using an access token.
  - OAuth2 is a standard for authorization which provides secure delegated access.
  - Application can access resources on a resource server, without sharing user credentials with application.
  - Tokens to be issued by an identity provider to the client applications, with the consent of the user. The client then uses the token to access the resource server.
- **OpenID** is a protocol for authentication.
  - User must obtain an openID account using OpenID identity provider. The user can use that openID account to sign into other web sites.
  - It has its support for ad-hoc client registration using a DH exchange and a way to verify assertions without making another round-trip to the provider.
- OpenID and **OpenID Connect** are both for *authentication*, not for *authorization*.
- **OpenID Connect** is an extension over OAuth 2.0 authorization framework.
  - Allows authentication and authorization.
  - Issue ID token - contains end user details in JSON Web Token (JWT) format.
  - Thus the ID token receiving client can validate the token and authenticate the end user.
  - improved naming and structure of OpenID 2.0 fields and parameters in order to be easier to use. OpenID Connect defines how to send a signed and encrypted request object where OAuth2 does not.

# Cont.

	SAML 2.0	OAuth2	OpenID Connect
<b>What is it?</b>	Open standard for authorization and authentication	Open standard for authorization	Open standard for authentication
<b>History</b>	Developed by OASIS in 2001	Developed by Twitter and Google in 2006	Developed by the OpenID Foundation in 2014
<b>Primary use case</b>	SSO for enterprise apps	API authorization	SSO for consumer apps
<b>Format</b>	XML	JSON	JSON

SAML 2.0, OpenID 2.0		OAuth 2.0	OpenID Connect
	Initiating user's login session		 Initiating user's login session
	Not responsible for collecting user consent		 Collecting user's consent to share attributes
	High-security identity tokens (SAML only)		 High-security identity tokens (using JSON Web Tokens)