Chapter 5

Identity and Access Management

Overview

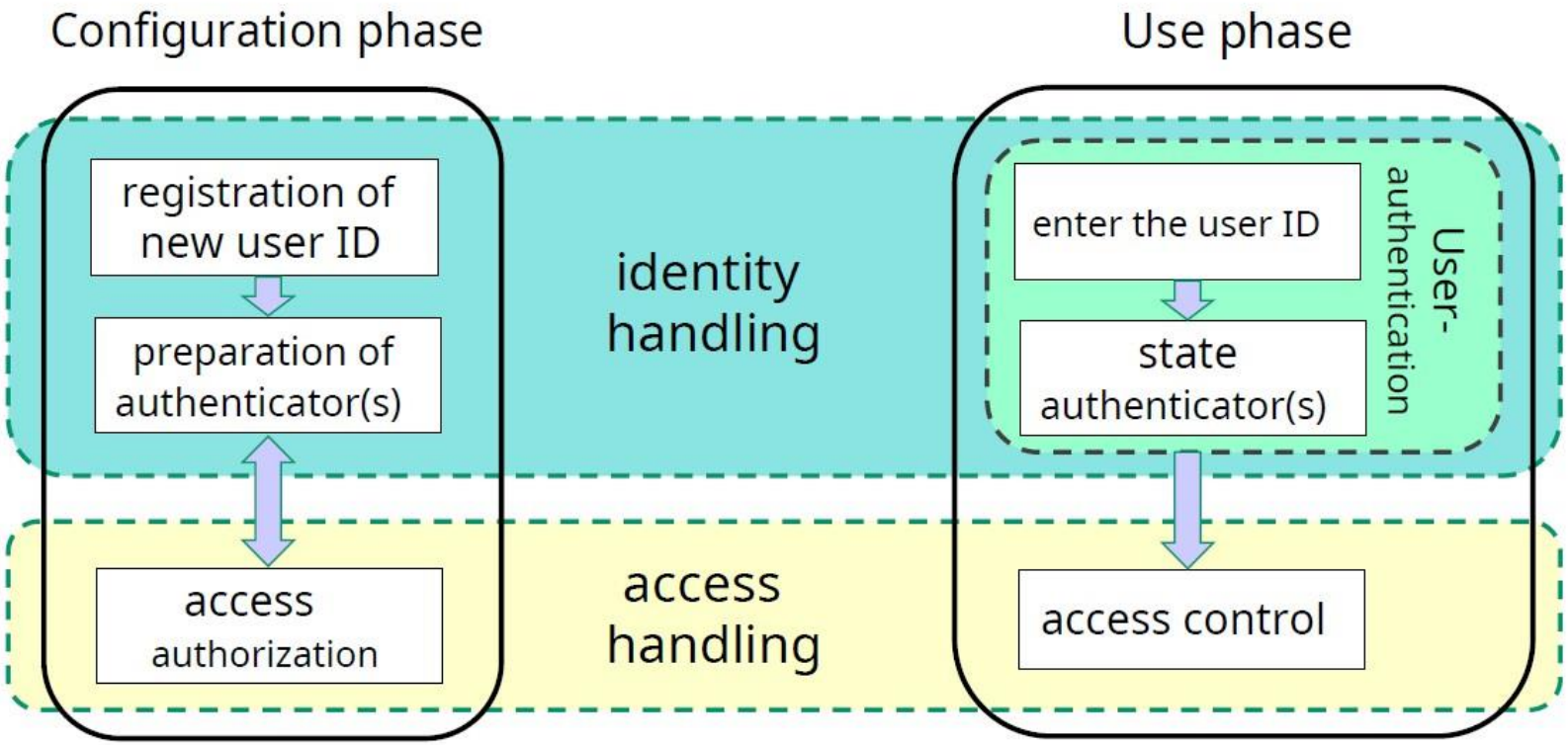
- What is IAM (Identity and Access Management)
- Identity and access management
- What is identity
- The silo model for identity management
- Federated model for identity management
  - User authentication as a service
- Access control

IAM-Definition

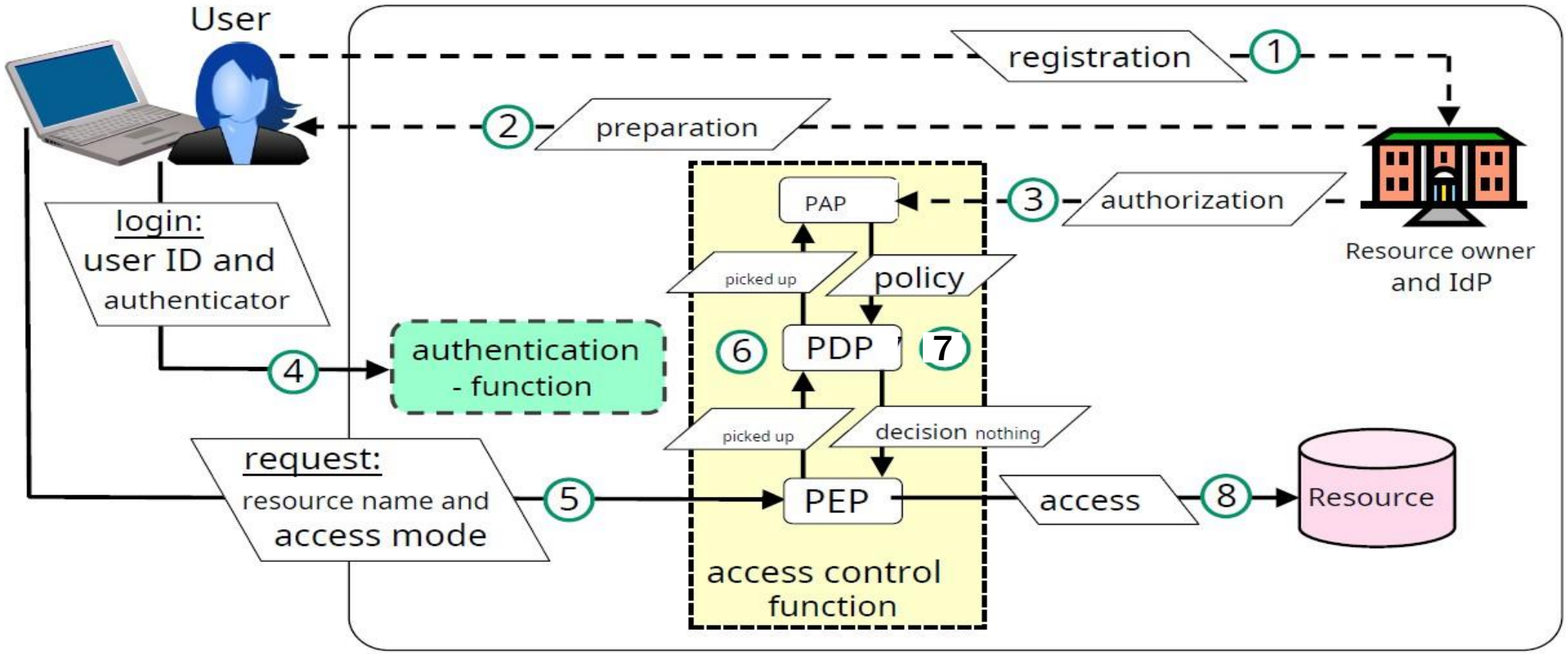
Identity and access management (IAM) is the security discipline that enables the right individuals to access the right resources at the right times for the right reasons.

Gartner, security glossary https://www.gartner.com/en/information-technology/glossary/identity-and-access-management-iam

Identity and Access Management



IAM scenario



PAP: Policy Administration Point

PDP: Policy Decision Point

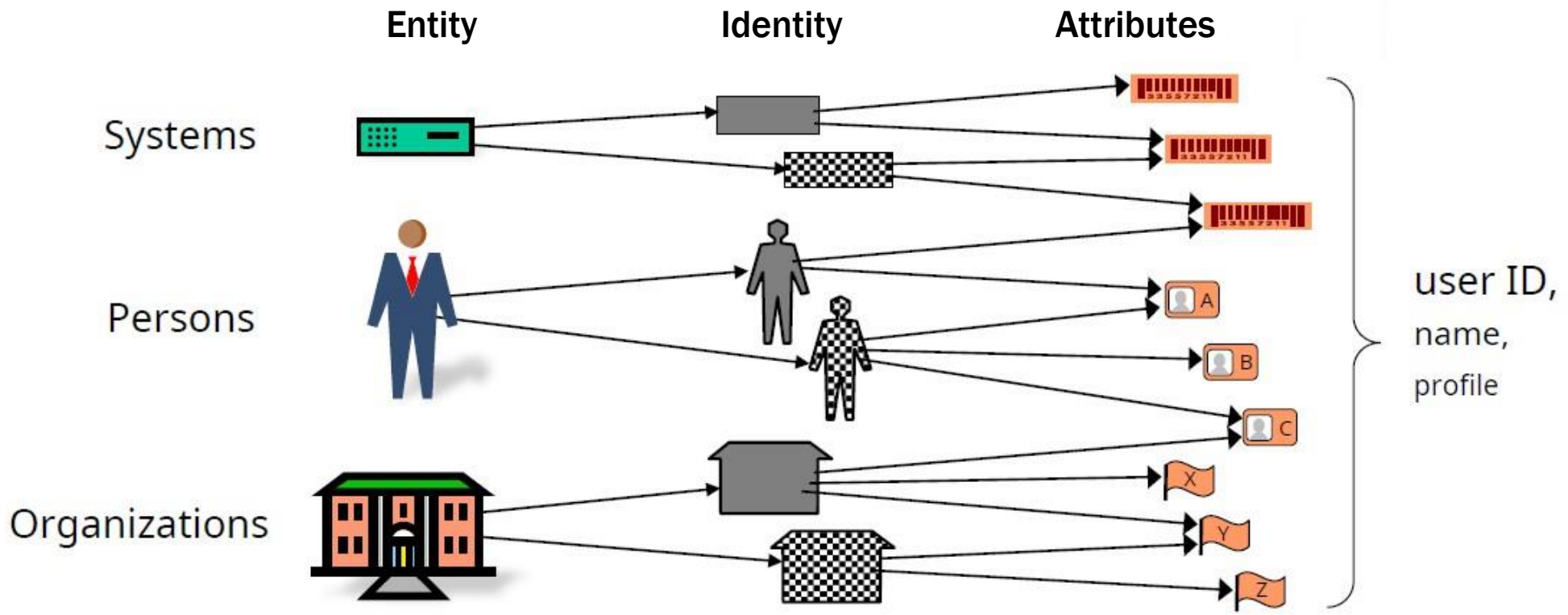
PEP: Policy Enforcement Point

IdP: Identity Provider

configuration phase

use phase

Identity Concept



Concepts related to identity

- Entity
  - A person, organization, agent, system, session, process, etc.
- Identity
  - A set of names / attributes for the entity in a particular domain
  - An entity can have identities in several domains
  - An entity can have several identities in the same domain
- Digital identity
  - Digital representation of names / attributes in a way that is suitable for digital processing
- Name and attribute of entities, can be
  - unambiguous or ambiguous within a domain
  - short-term or permanent,
  - self-defined or defined by authority,
  - processed by humans and/or computers
  - etc.
- An identifier is a uniquely unique name

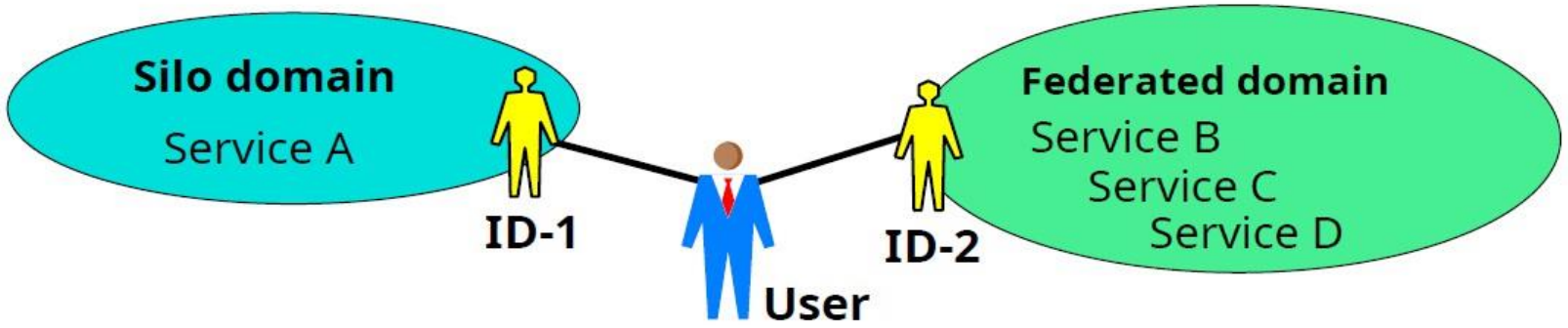
Identity - creation - registration

- Authentication requires that an identity has been registered

- Registration can take place in two ways:
  - pre-authentication, from previous identity, e.g. passport
  - creation of a new identity

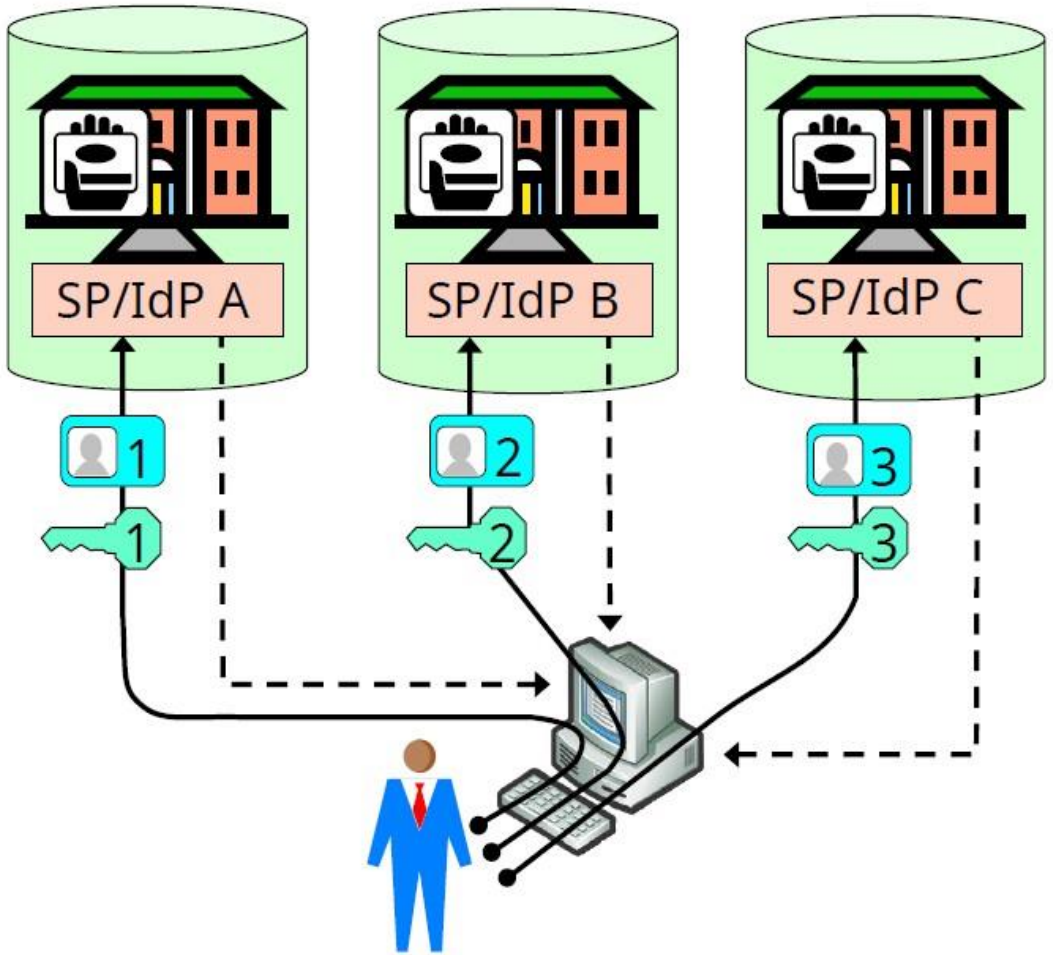
Identity Domains

- An identity domain has a namespace with unique names
  - The same user can have separate identities in different domains
  - The same user normally only has one identity in a domain, but it is entirely possible that the same user can have several identities in a domain.



- Silo domains with one authority, e.g. corporate network
- Federated identity domains
  - The identity domain can be used by many different service providers
  - Requires cooperation on identity policy between service providers

Silo identity management



Silo domains

Explanation:

SP (Service Provider)

IdP (Authentication Server)

Silo identity domain

X User ID

X Authenticator

Login/request

Service

Silo identity management

- SP (Service Provider) = IdP (Authentication Server):
  - SP manages the namespace and performs authentication
- Uniquely unique identifiers assigned to each user
- Benefits
  - Easy to set up, low initial costs
  - Potentially good basis for strong privacy protection
- Cons
  - Identity overload for users, poor user-friendliness, poor integration of services between providers
  - Low acceptance of new services with a separate ID and authenticator
  - Users must provide the same information to many service providers
  – For service providers: Barrier against the collection of user data

Actors in federated identity management

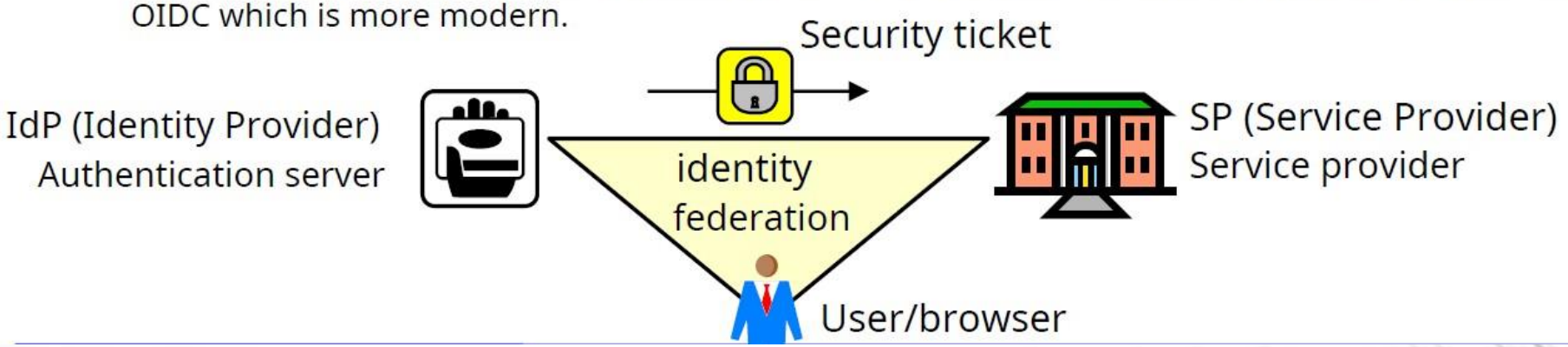
- User
  - Has user ID and authenticator(s)
  - Want to use services from different SPs.

- Service provider (SP) (Service Provider)
  - Has a register of user IDs
  - Has an agreement with one or more IdPs for user authentication

- Authentication server (IdP) (Identity Provider)
  - Has a register of user IDs
  - Has an agreement with SPs
  - Provides/sells user authentication as a service to SPs

Protocols / standards for ID federation

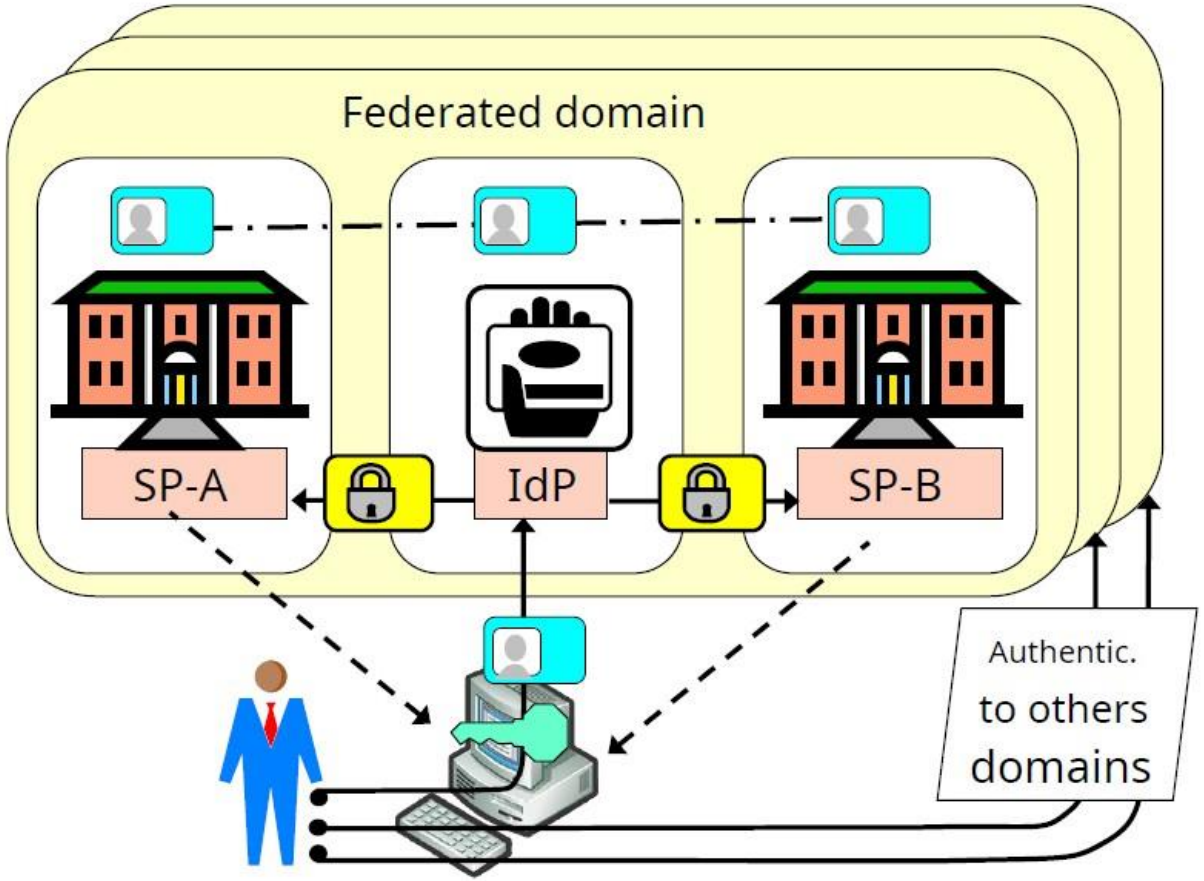
- Involves several entities
  - User/browser, IdP, SP, and sometimes a "broker" (e.g. the ID gateway)
- The IdP authenticates the user, generates and sends a digitally signed security ticket to the SP.
- SP receives a security ticket (Security Assertion) as proof that the user has been authenticated.
- Standards:
  - OAuth (Open Authorization)
  - OIDC (OpenID Connect), which is based on OAuth
  - SAML (Security Assertions Markup Language),
    - SAML was a previously widespread standard for ID federation, but approx. By 2020, everyone had switched to OIDC which is more modern.



ID federation - pros and cons

- Benefits
  - better user-friendliness
  - Allows SPs to focus on services, not having to deal with authenticators
  - Allows IdPs to collect information about user usage patterns
  - Scales well within a sector, provides good quality in authentication
- Cons
  - Technical and legal complexity
  - Trust requirements between actors
    - Every actor can potentially compromise security
  - Issues for privacy,
    - Massive collection/exchange of data between SP and IdP is a threat to privacy
  - Limited scalability between different sectors/domains,
    - Limited by political and economic constraints
    - A federated domain can become a new form of silo in relation to other domains

ID federation use-case



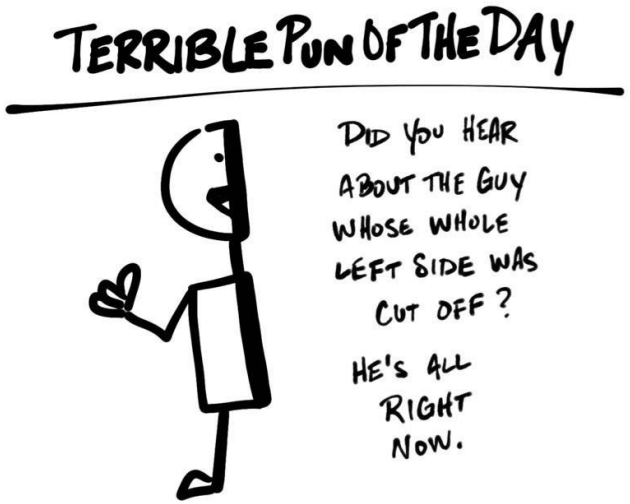
Explanation:

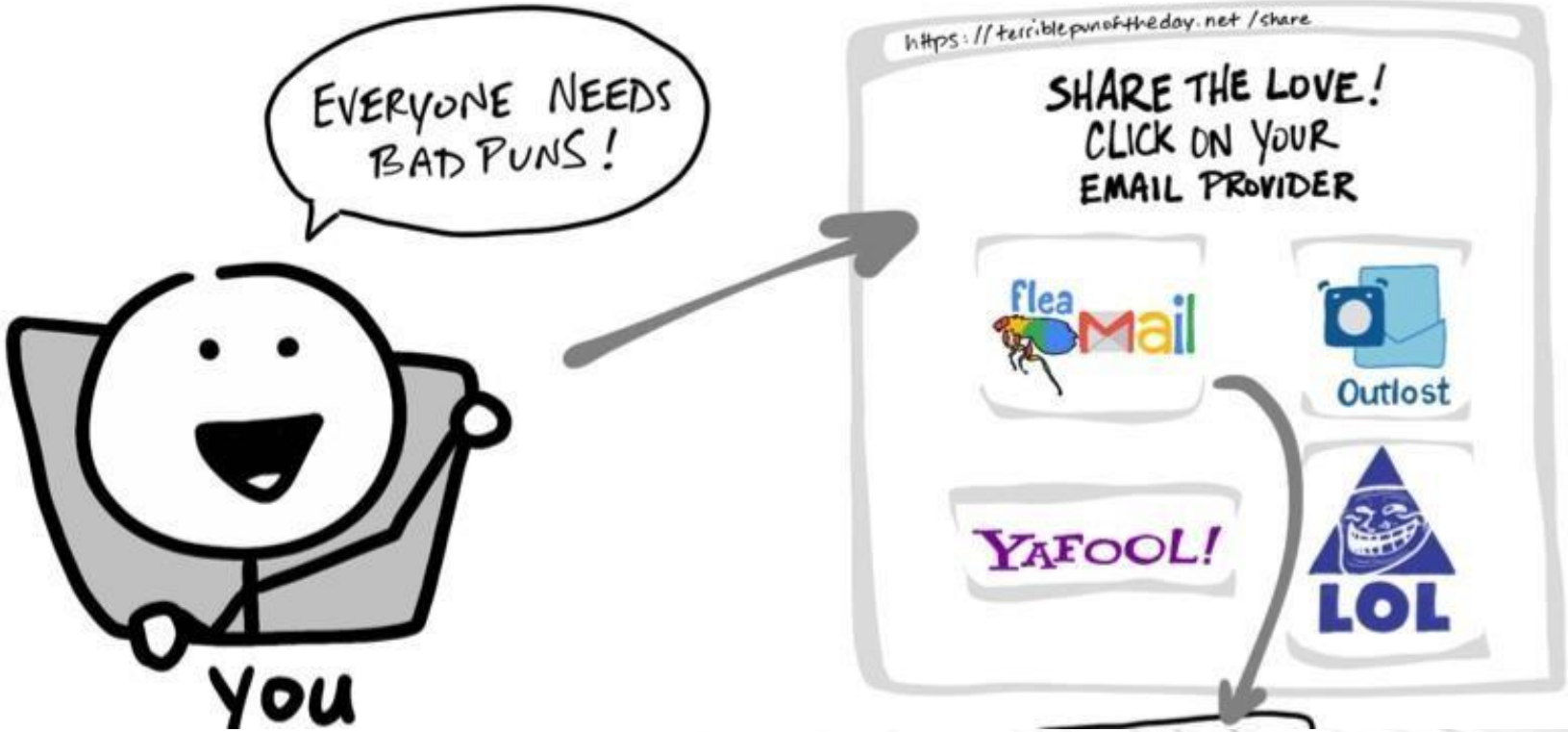
- SP (Service Provider)
- IdP (Authentication Server)
- Identity domain
- User ID
- Authenticator
- Security ticket
- Login
- Service
- Federated Id

OAUTH2.0

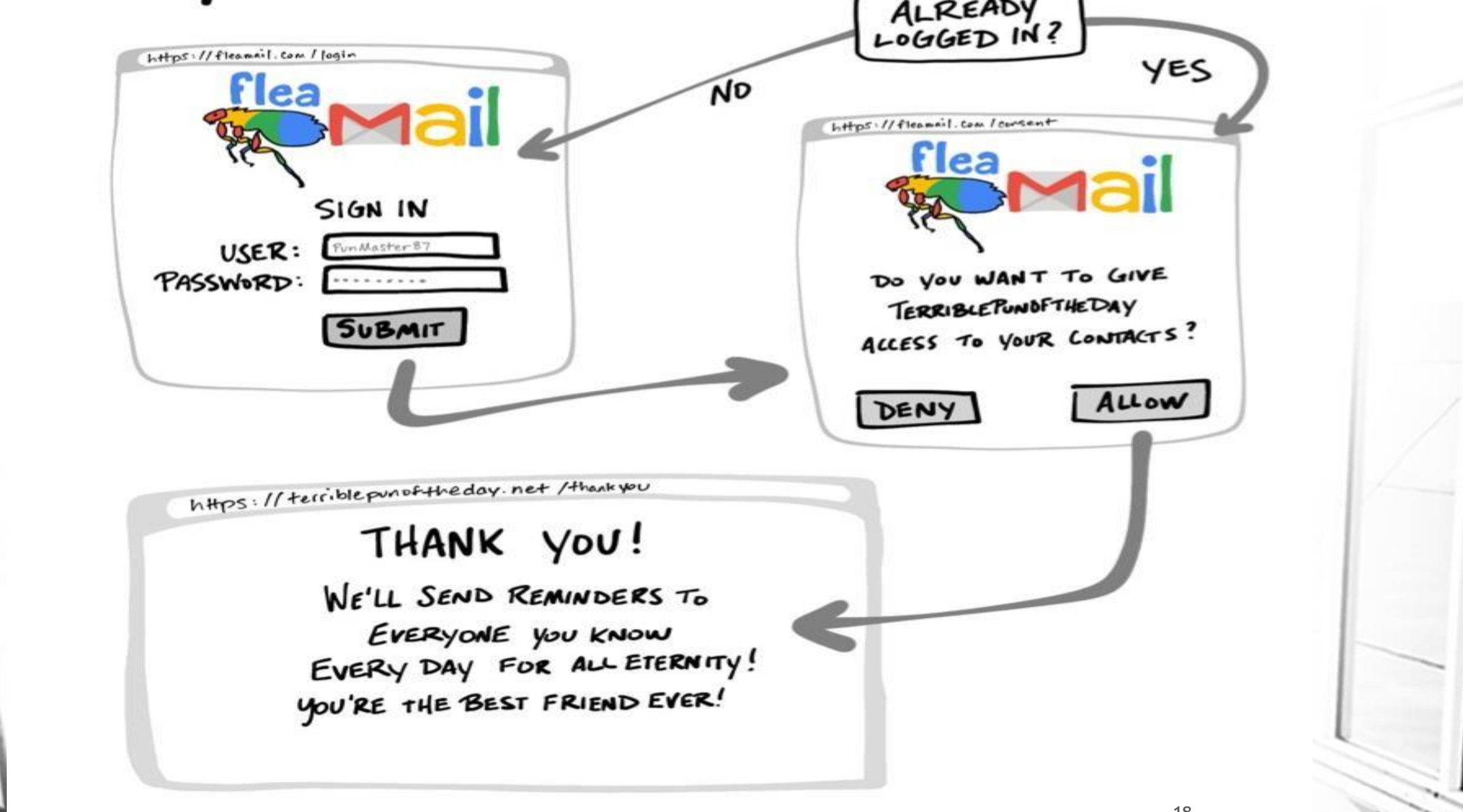
- It stands for “Open Authorization”, is a standard designed to allow a website or application to access resources hosted by other web apps on behalf of a user.
- OAuth 2.0 is an authorization protocol and NOT an authentication protocol. As such, it is designed primarily as a means of granting access to a set of resources, for example, remote APIs or user data.
- OAuth 2.0 uses Access Tokens. An Access Token is a piece of data that represents the authorization to access resources on behalf of the end-user.
- Although the web is the main platform for OAuth 2, the specification also describes how to handle this kind of delegated access to other client types (browser-based applications, server-side web applications, native/mobile apps, connected devices, etc.)

How OAUTH works?





https://developer.okta.com/blog/2019/10/21/illustrated-guide-to-oauth-and-oidc


ALREADY LOGGED IN?
NO
YES
https://fleamail.com/login
flea mail
SIGN IN
USER:
PunMaster87
PASSWORD:
SUBMIT
https://fleamail.com/consent
flea mail
DO YOU WANT TO GIVE TERRIBLEPUNOFTHEDAY ACCESS TO YOUR CONTACTS?
DENY
ALLOW
https://terriblepunoftheday.net/thankyou
THANK YOU!
WE'LL SEND REMINDERS TO EVERYONE YOU KNOW EVERY DAY FOR ALL ETERNITY!
YOU'RE THE BEST FRIEND EVER!

OAuth terminologies

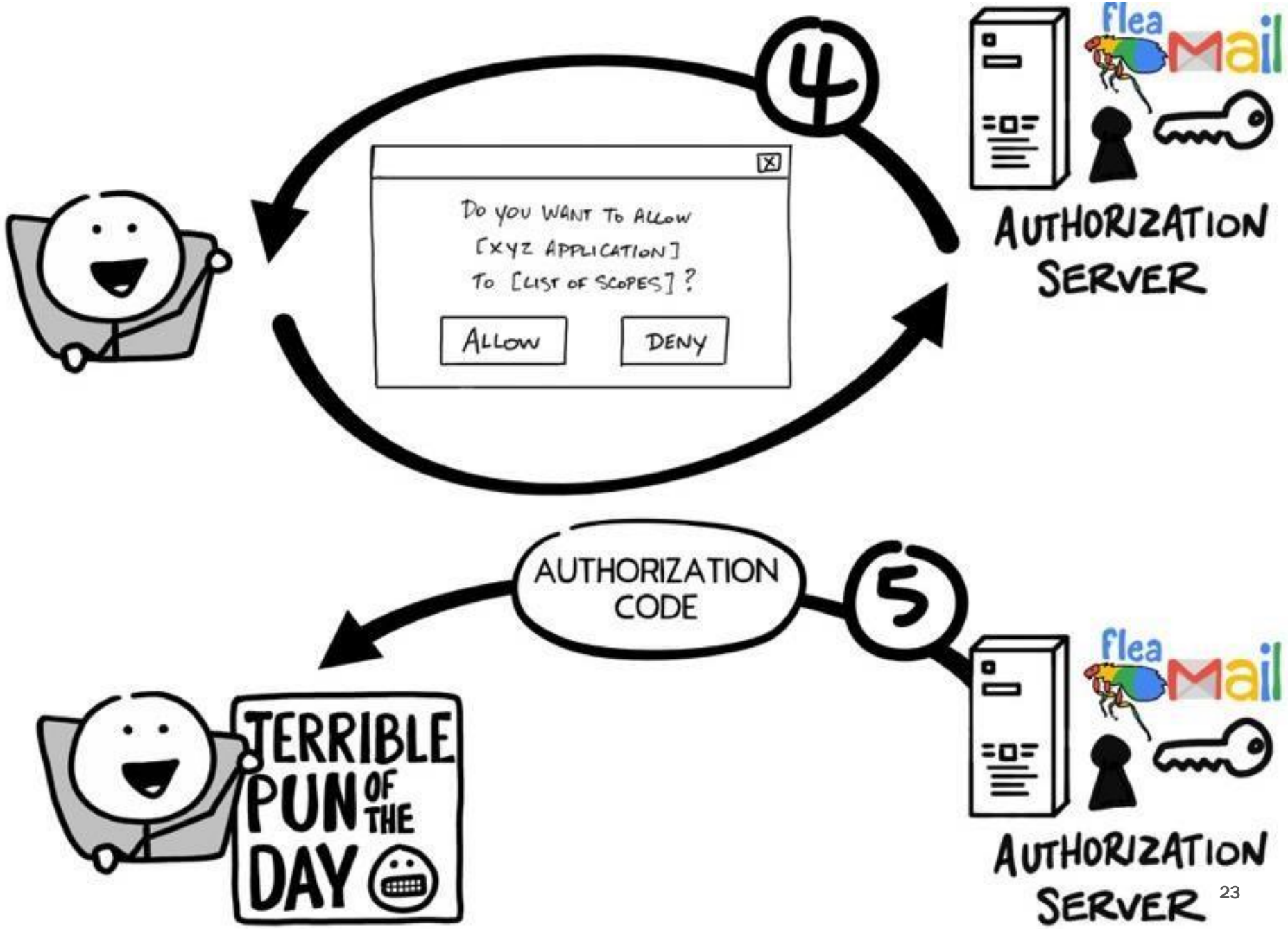
- Resource Owner: You! You are the owner of your identity, your data, and any actions that can be performed with your accounts.
- Client: The application (e.g. “Terrible Pun of the Day”) that wants to access data or perform actions on behalf of the Resource Owner.
- Authorization Server: The application that knows the Resource Owner, where the Resource Owner already has an account.
- Resource Server: The Application Programming Interface (API) or service the Client wants to use on behalf of the Resource Owner.
- Redirect URI: The URL the Authorization Server will redirect the Resource Owner back to after granting permission to the Client. This is sometimes referred to as the “Callback URL.”

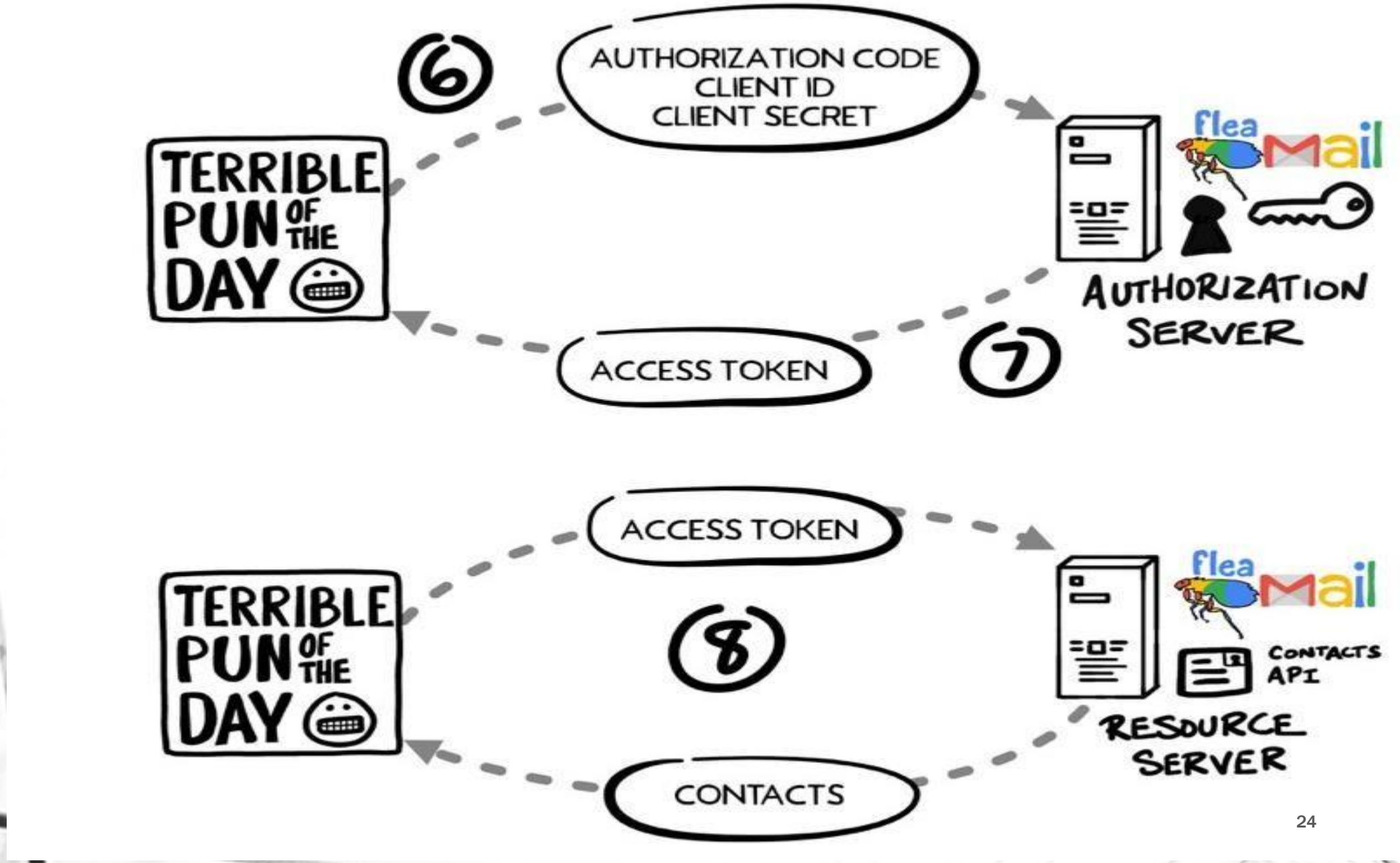
- **Response Type**: The type of information the Client expects to receive. The most common Response Type is code, where the Client expects an Authorization Code.
- **Scope**: These are the granular permissions the Client wants, such as access to data or to perform actions.
- **Consent**: The Authorization Server takes the Scopes the Client is requesting, and verifies with the Resource Owner whether or not they want to give the Client permission.
- **Client ID**: This ID is used to identify the Client with the Authorization Server.
- **Client Secret**: This is a secret password that only the Client and Authorization Server know. This allows them to securely share information privately behind the scenes.

- **Authorization Code:** A short-lived temporary code the Client gives the Authorization Server in exchange for an Access Token.

- **Access Token:** The key the client will use to communicate with the Resource Server. This is like a badge or key card that gives the Client permission to request data or perform actions with the Resource Server on your behalf.

OAUTH 2.0 Flow


4
DO YOU WANT TO ALLOW
[XYZ APPLICATION]
TO [LIST OF SCOPES]?
ALLOW
DENY
AUTHORIZATION SERVER
AUTHORIZATION CODE
5
TERRIBLE PUN OF THE DAY
AUTHORIZATION SERVER


6
AUTHORIZATION CODE
CLIENT ID
CLIENT SECRET
TERRIBLE PUN OF THE DAY
flea Mail
AUTHORIZATION SERVER
ACCESS TOKEN
7
ACCESS TOKEN
8
TERRIBLE PUN OF THE DAY
flea Mail
CONTACTS API
RESOURCE SERVER
CONTACTS

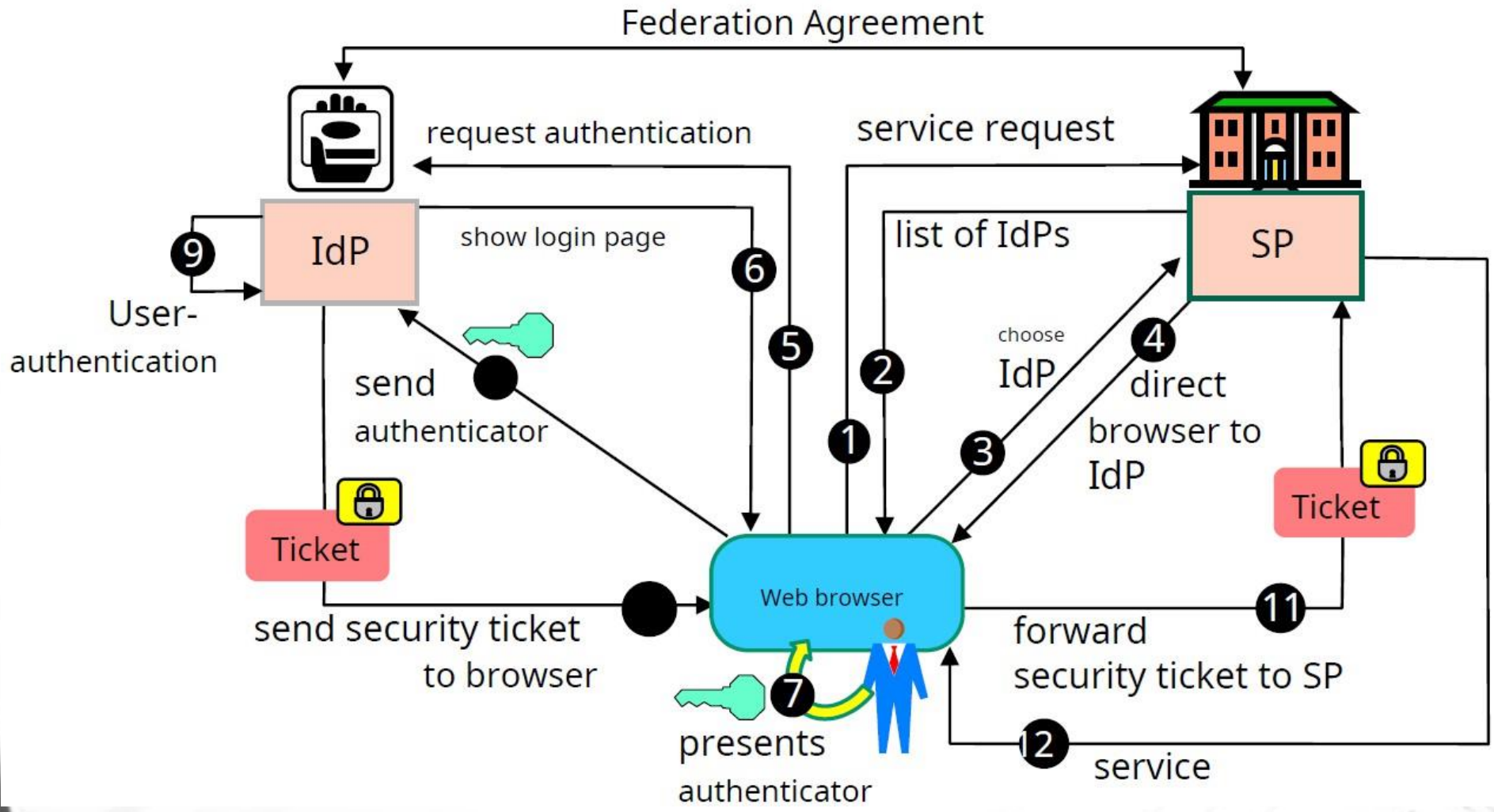
Explained Flow

- You, the Resource Owner, want to allow “Terrible Pun of the Day,” the Client, to access your contacts so they can send invitations to all your friends.
- The Client redirects your browser to the Authorization Server and includes with the request the Client ID, Redirect URI, Response Type, and one or more Scopes it needs.
- The Authorization Server verifies who you are, and if necessary prompts for a login.
- The Authorization Server presents you with a Consent form based on the Scopes requested by the Client. You grant (or deny) permission.
- The Authorization Server redirects back to Client using the Redirect URI along with an Authorization Code.
- The Client contacts the Authorization Server directly (does not use the Resource Owner’s browser) and securely sends its Client ID, Client Secret, and the Authorization Code.
- The Authorization Server verifies the data and responds with an Access Token.
- The Client can now use the Access Token to send requests to the Resource Server for your contacts.

Open ID Connect (OIDC

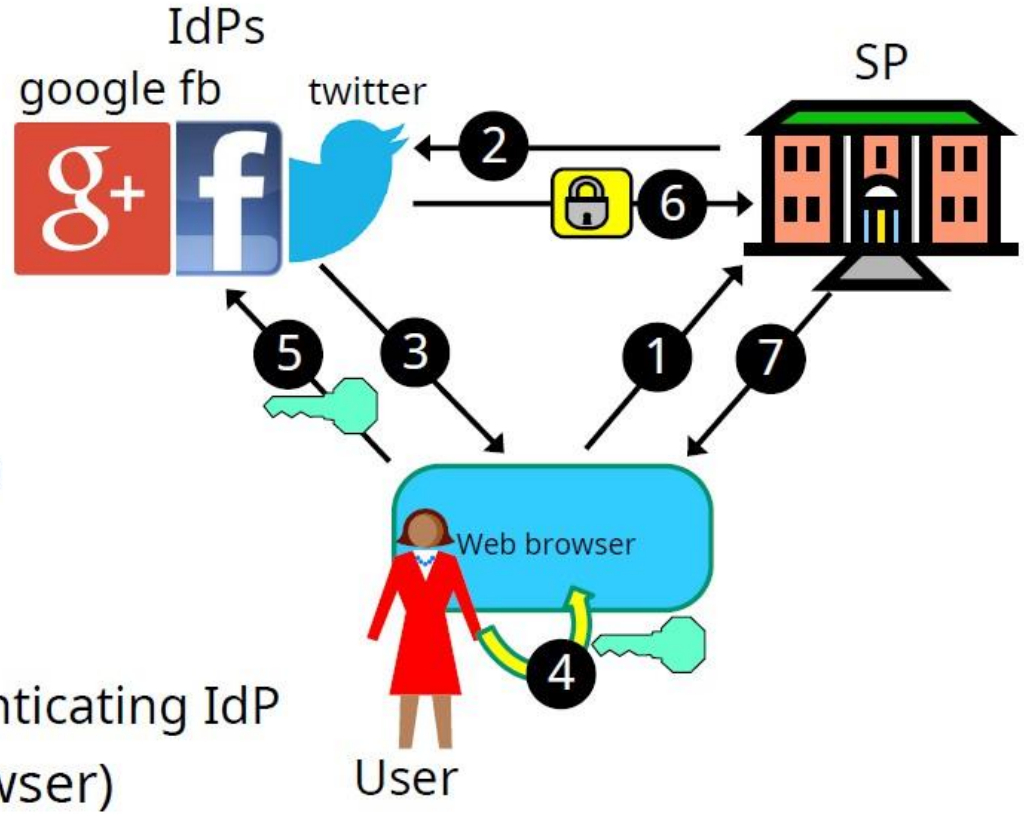
- OIDC is based on the OAuth 2.0 specification. It is a thin layer that sits on top of OAuth 2.0 that adds login and profile information about the person who is logged in.
- SPs establish federation agreements with IdPs through OAuth - e.g. Airbnb (SP) with Facebook (IdP)
- OpenID Connect enables scenarios where one login can be used across multiple applications, also known as *single sign-on* (SSO). For example, an application could support SSO with social networking services such as Facebook or Twitter so that users can choose to leverage a login they already have and are comfortable using.

OIDC authentication



Identity federation with google, facebook and twitter

authentication with the IdPs
google, facebook or twitter

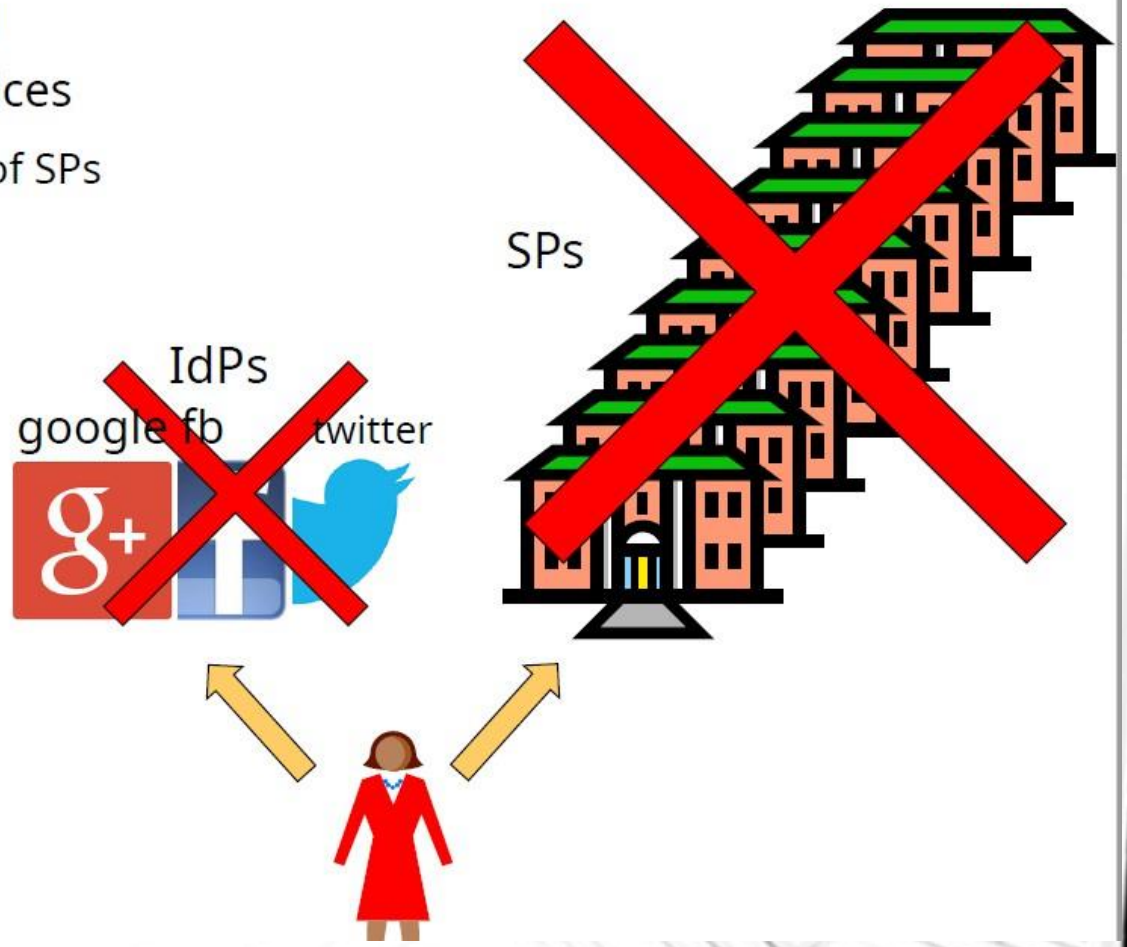


Simplified IODC scenario

1. User requests service
2. Redirect browser to selected IdP for authentication
3. Present login page from IdP
4. User provides ID and authenticator(s)
5. User ID and authenticator are sent to the authenticating IdP
6. Security ticket is sent to SP (actually via browser)
7. SP provides service to user

IdP dependency

- Facebook went down for 6 hours on 4 October 2021
  - Millions of users could not log on to online services
- The number of IdPs is very low compared to the number of SPs
- IdP dependency is a problem
- Need for alternative login



End of lecture