Open ID Connect（中途半端なまとめです）

●OAuth2.0 と似ているが、アクセストークンではなく、ID トークン（ユーザ ID や PW 等が含まれる）によって、リソースサーバは認可ではなく、クライアントを介してのユーザ認証を行う。つまり、アクセストークンではユーザ情報が不明であったが、ID トークンからは明確になる。

●ID トークンは、認可サーバではなく、OpenID プロバイダ（OP）から発行されるが、OP に認可サーバの機能を付加することによって、アクセストークンの同時発行ができる。これによって、認証及び認可を経てクライアントはリソースサーバとのデータの受け渡しができる。

※というより、クライアントがユーザと OP 間で認可手続きをさせようとしたら、その前に認証手続きが始まって、その後、付与する権限が決まる流れと考えたほうがいい。あとは、クライアントが認可コードを使って、OP に ID トークンとアクセストークンを要求する流れになる。

●ここから、根拠が怪しくなるが、クライアントを経て ID トークンをリソースサーバに送るため、クライアントが変わっても、ユーザが変わらなければ、ID トークンは変わらないので、ID や PW は共通のままとなる。

PKCE（Proof Key for Code Exchange）

●OAuth2.0 の拡張仕様であり、認可コード横取り攻撃対策

●例として、code_challenge_method の値が S256 であるといわれた場合の意味は、**「SHA256 でハッシュ化したものを base64url でエンコードするアルゴリズム」**である。

●クライアントから認可サーバへ送る認可コード要求リクエスト（クライアントが Web ブラウザ経由で認可サーバへリダイレクト）の際に、**「code_challenge_method の値が S256」**という情報と S256 を適用させた値**「B:チャレンジコード」**を送る。

●その後、クライアントから、認可コードを免許として、認可サーバへ送るアクセストークン要求リクエストに、S256 を適用させる前の値**「A:ランダムな検証コード」**を送る。

●そして、認可サーバにおいて、A に対して S256 を適用させた値と B が一致すれば、中間者攻撃を受けていないことを確認することができる。

●上記の横取り攻撃対策は、認可サーバが検証しているが、どの機器が検証や確認をするのかは、問題文によって変わってくるため、先入観を捨てる

FIDO（Fast IDentity Online）

●FIDO アライアンスによるパスワードレス認証を実現した認証方式の規格である。

●FIDO UAF は、FIDO クライアントと認証器は同一であり、スマホ等である。認証サーバ（FIDO サーバ）からのチャレンジコードに対して認証器がデジタル署名を施す処理等はご存じのとおり。

●FIDO 2 は、FIDO クライアントは Web ブラウザ、認証器はスマホ等で分割されており、この 2 つを繋げているのが Web ブラウザに備わっている Webauthn（Web Authentication：Web 認証 API)である。それ以外は FIDO UAF と同じ