AWS re:Invent

SEC409-R

Fine-grained access control for serverless apps

Eran Medan

Senior Security Consultant

Amazon Web Services

Agenda

Amazon Cognito overview

Demo

Architecture overview

Deep dive into the code

Amazon Cognito overview

The genesis of the next great application

Aspiring enterprise app creator

Supportive colleague

Hey, I've got an idea for a cool new internal app!

Nice, what are some of the cool features?

~~It will have Single Sign-On!~~

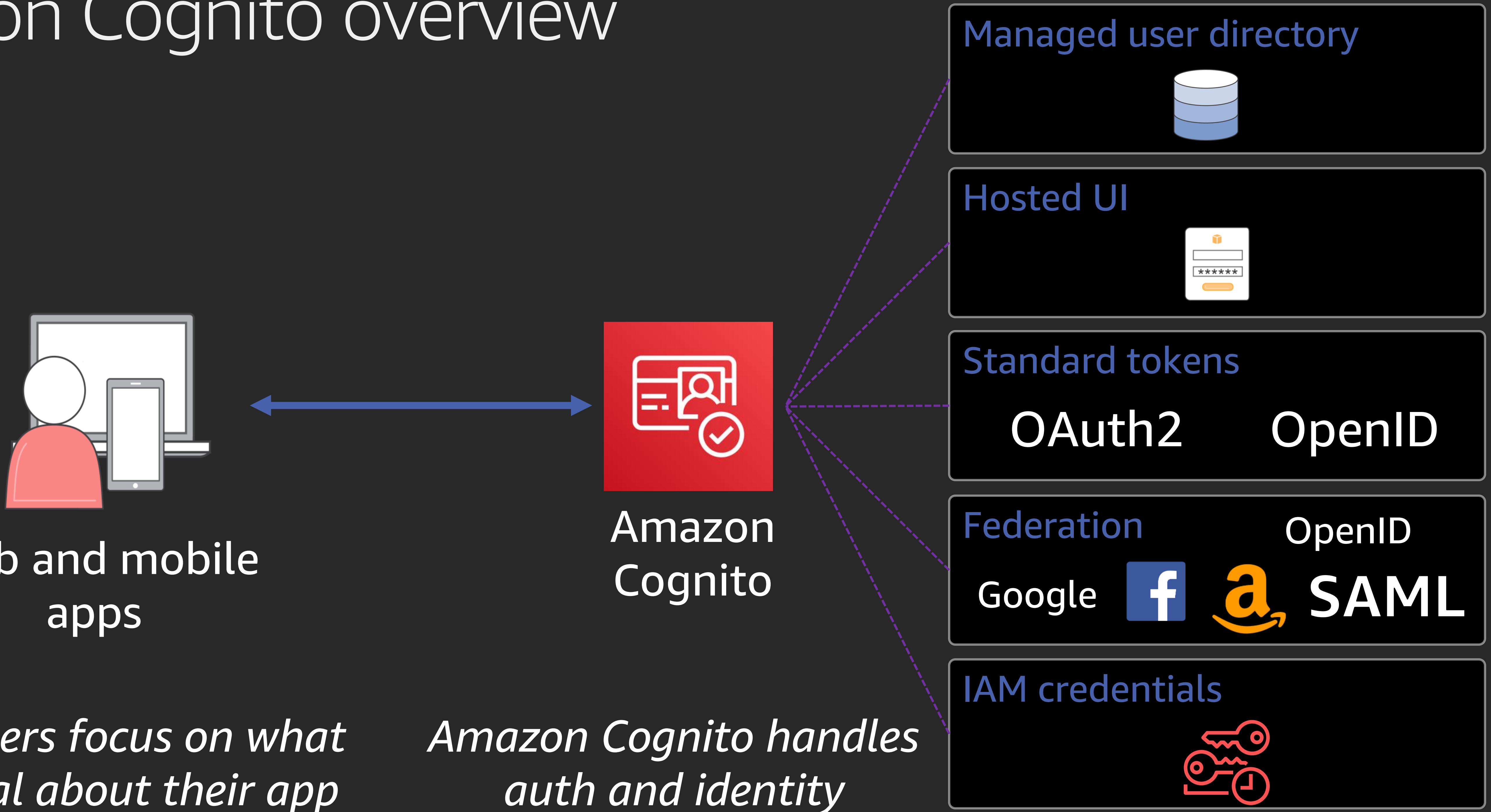
It will let you set up meetings in VR!

Where do I sign up?

Identity is undifferentiated heavy lifting

1. Need to develop a reliable user directory to manage identities
2. Handling user data and passwords and protecting privacy
3. Utmost requirements for availability and fault tolerance
4. Implementing token-based authentication
5. Supporting social identity providers
6. Federating with corporate directories

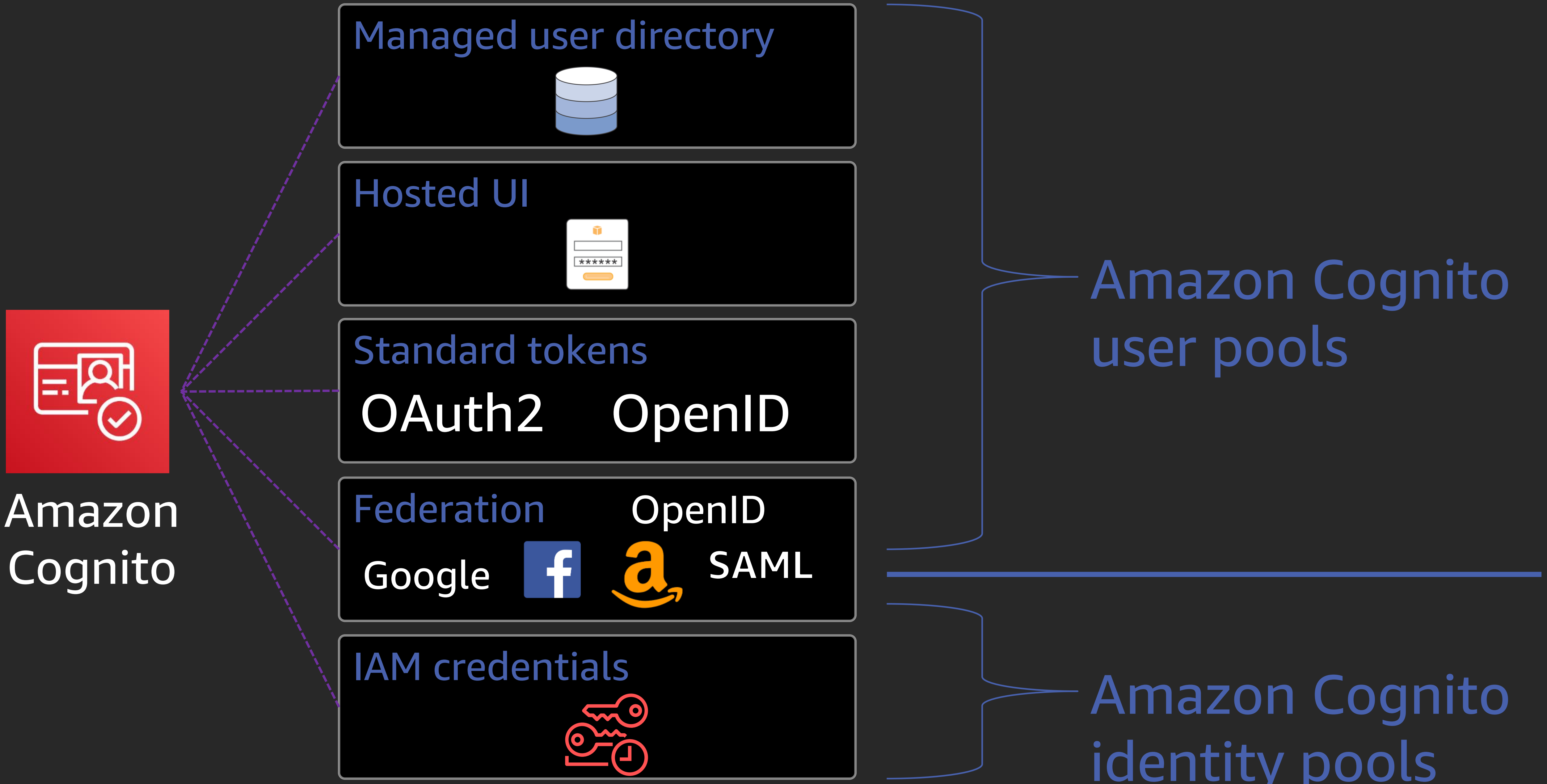
Amazon Cognito overview



Developers focus on what is special about their app

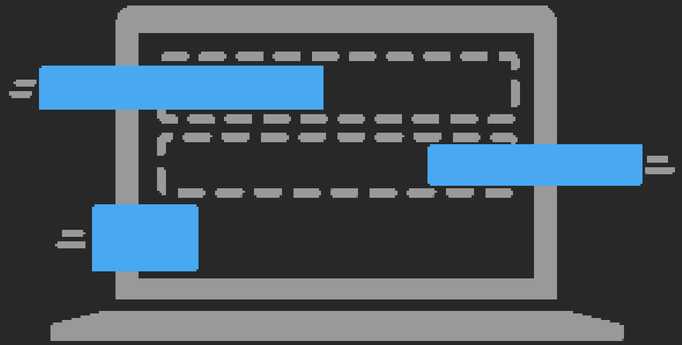
Amazon Cognito handles auth and identity

Amazon Cognito overview



Key features in Amazon Cognito user pools

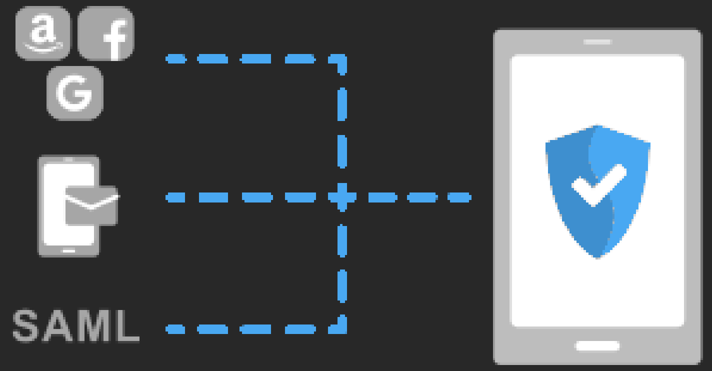
Secure, low-cost, and fully managed user directory that scales to 100s of millions of users



Built-in, customizable user interface for sign-up/sign-in

OAuth2

OAuth 2.0 support



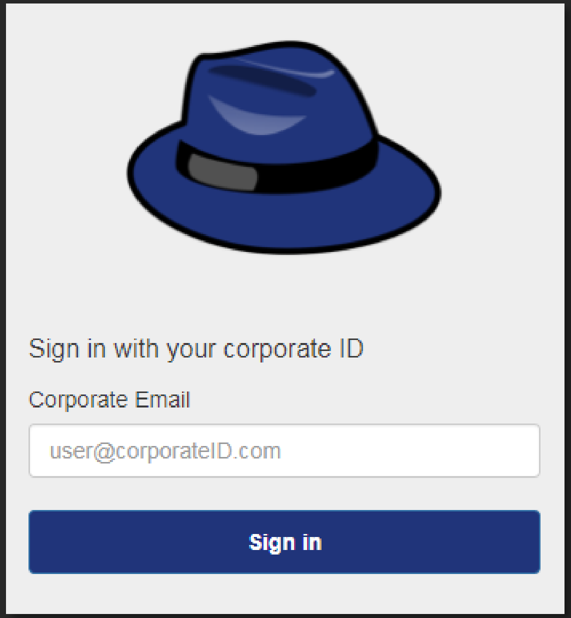
Federation with Facebook, login with Amazon, Google, and custom OIDC/SAML providers

Managed user directory

- Serverless directory
  - Nothing to manage
  - API driven
  - Multi-AZ redundancy
- User & group storage
  - Profile information (name, email, etc.)
  - Credential & device information (SRP verifier, MFA, etc.)
  - Extensible with custom attributes

Hosted user interface

- Facilitates user flows (sign up/in, forgot password, etc.)
- Customizable logo, style, and branding
- Use your own domain

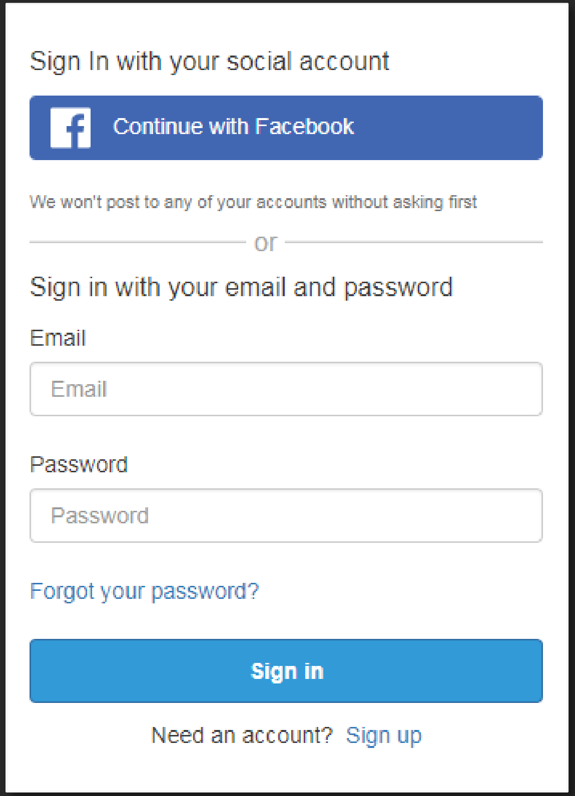


Sign in with your corporate ID

Corporate Email

user@corporateID.com

Sign in



Sign In with your social account

Continue with Facebook

We won't post to any of your accounts without asking first

or

Sign in with your email and password

Email

Email

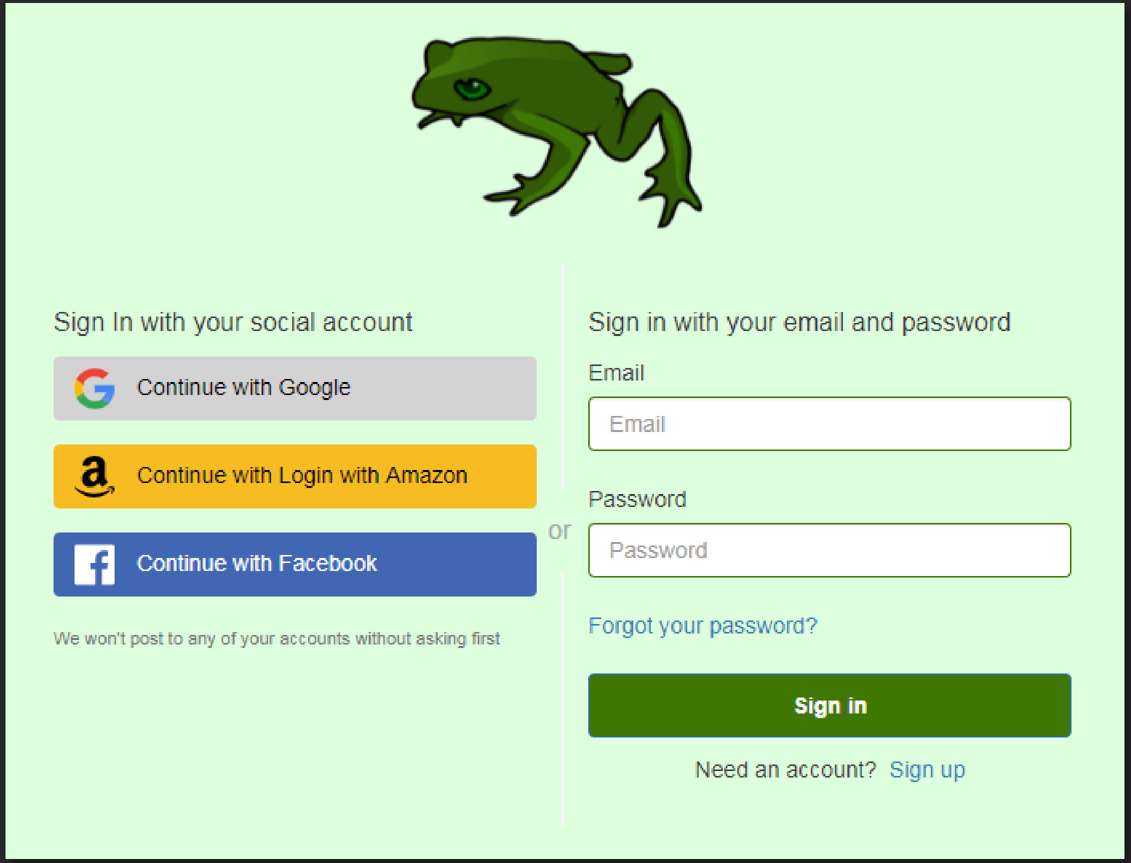
Password

Password

Forgot your password?

Sign in

Need an account? Sign up



Sign In with your social account

Continue with Google

Continue with Login with Amazon

Continue with Facebook

We won't post to any of your accounts without asking first

or

Sign in with your email and password

Email

Email

Password

Password

Forgot your password?

Sign in

Need an account? Sign up

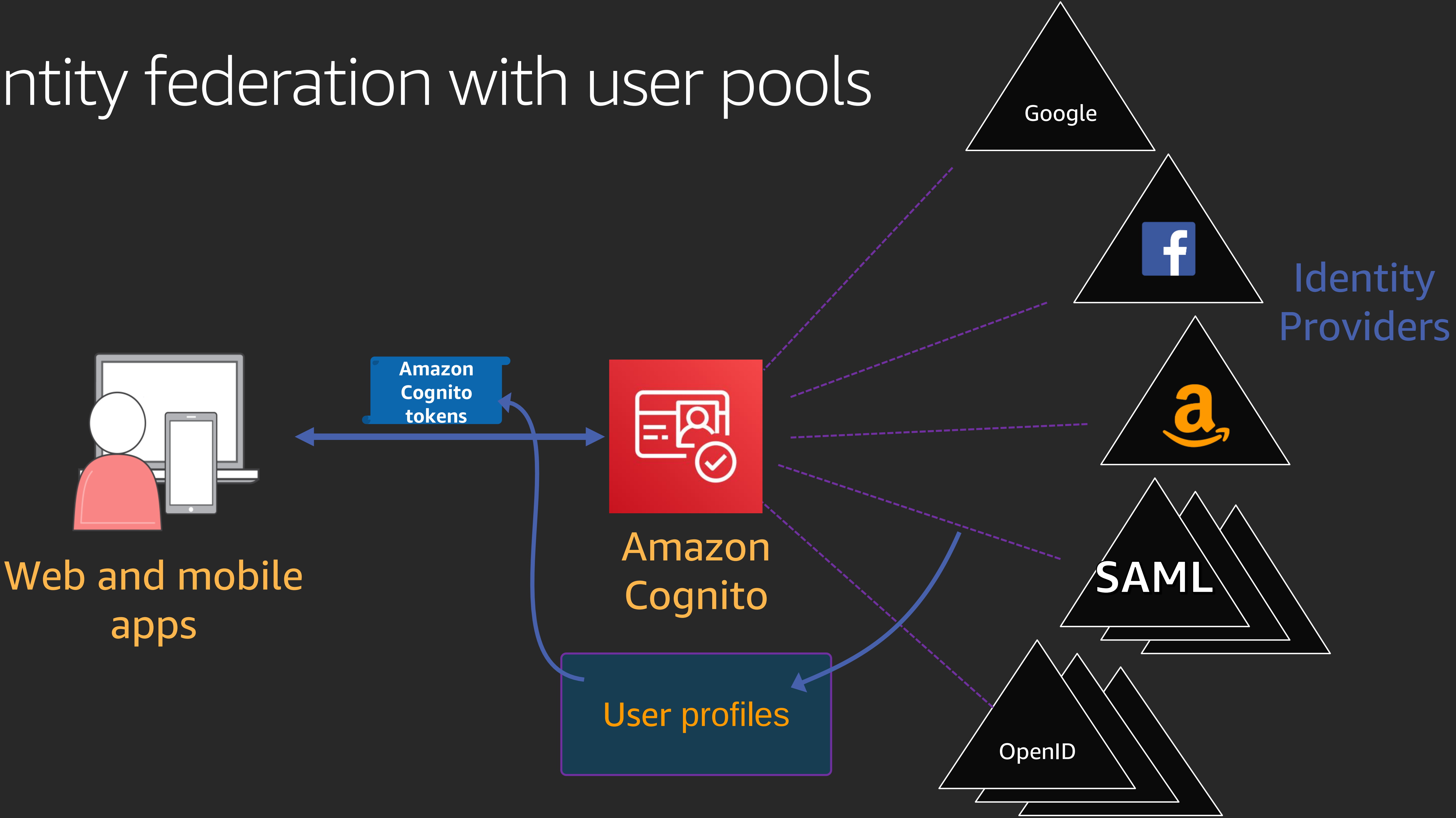
Support for OAuth 2.0 in Amazon Cognito user pools

- OAuth 2.0 flows:
  - Authorization code
  - Implicit
  - Client credentials

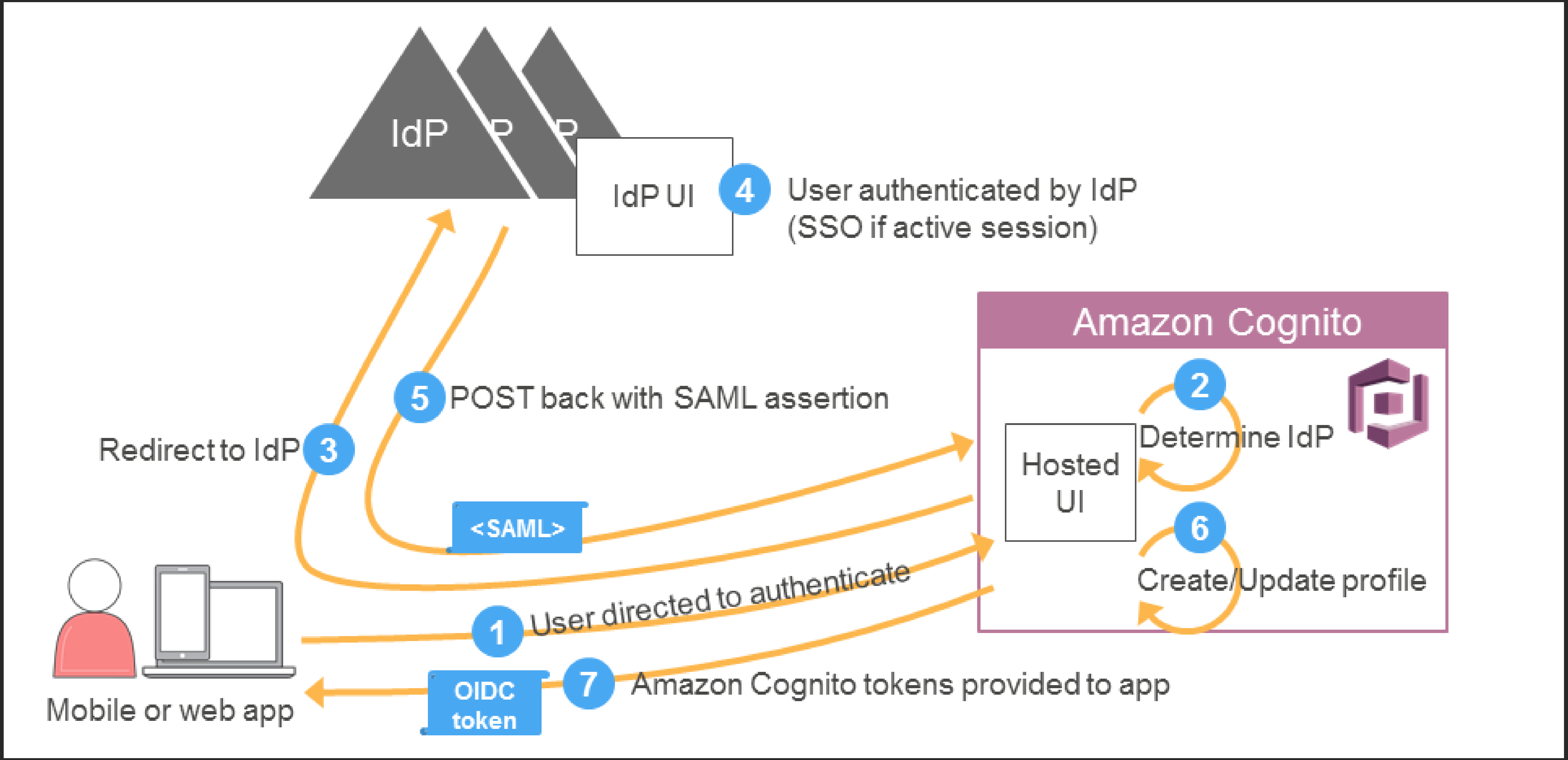
- Custom scopes defined for resource servers

Authorization Code Flow is now recommended also for browser based apps
https://tools.ietf.org/html/draft-ietf-oauth-browser-based-apps-01

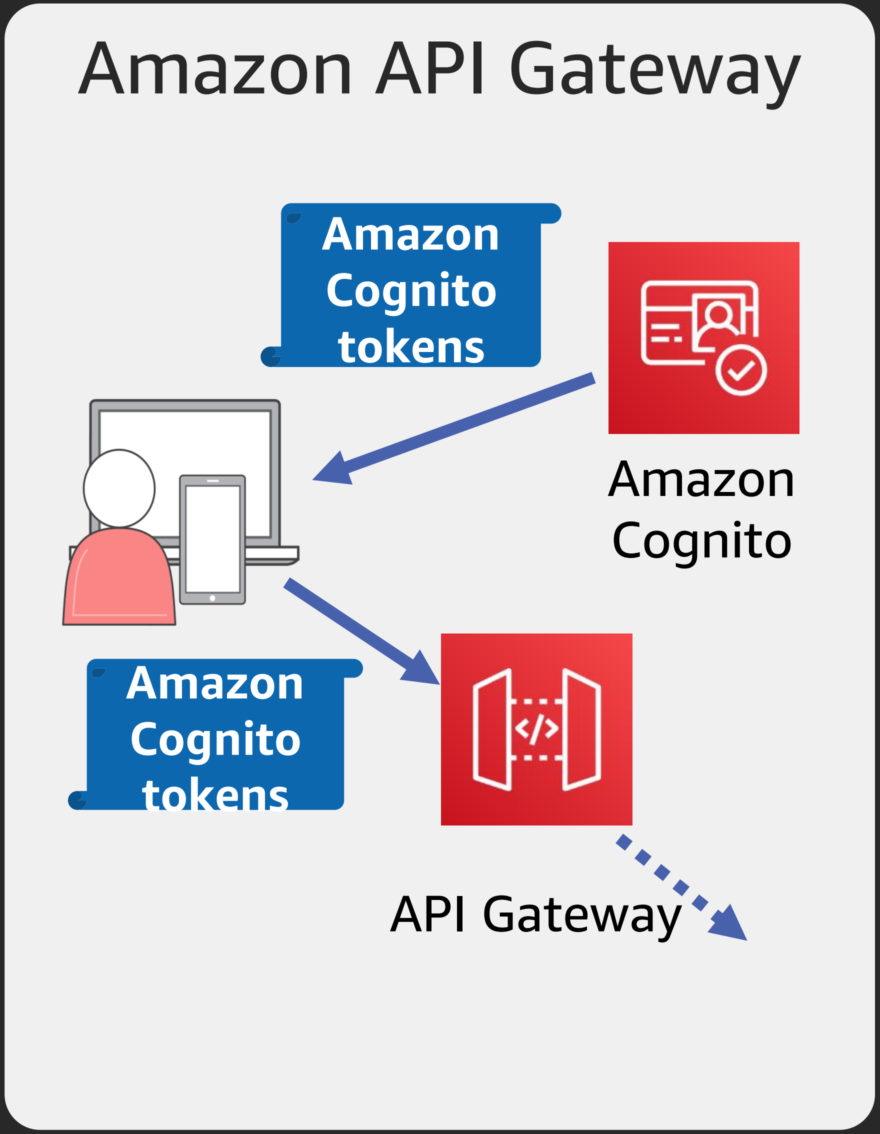
Identity federation with user pools

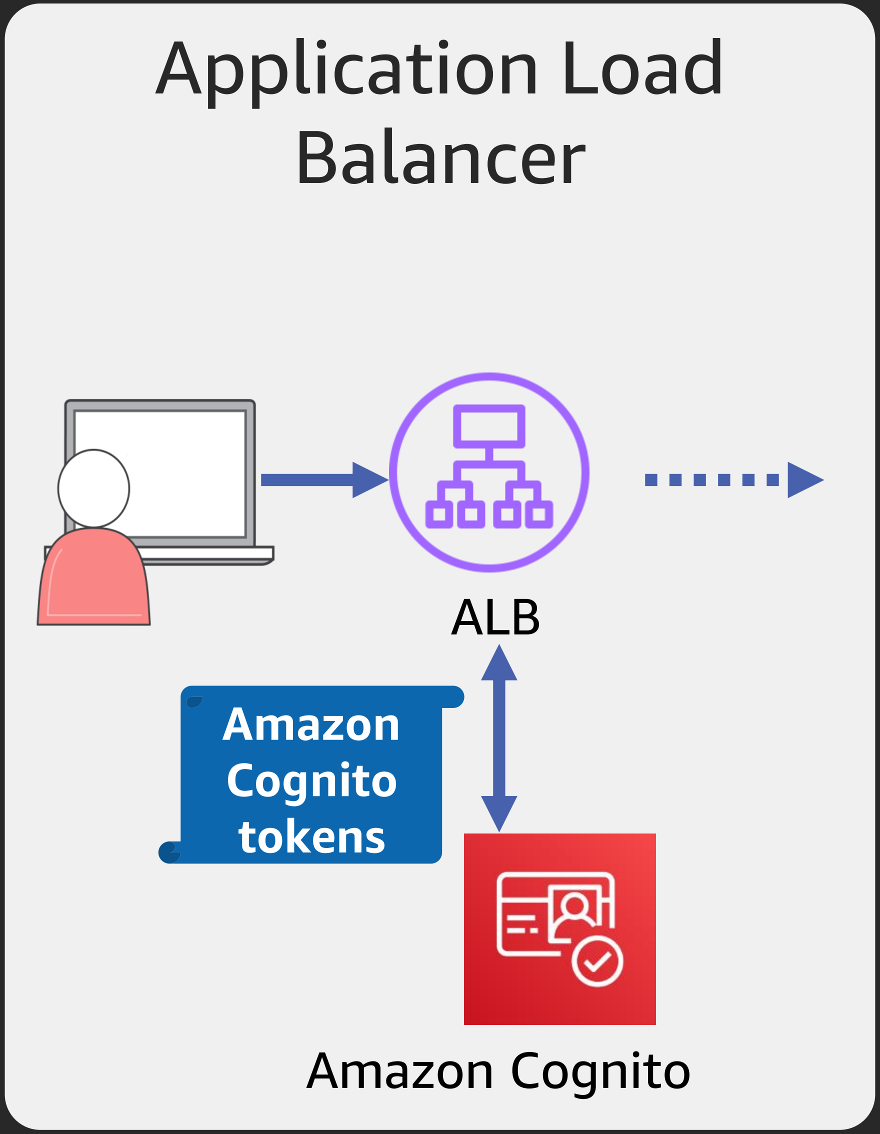


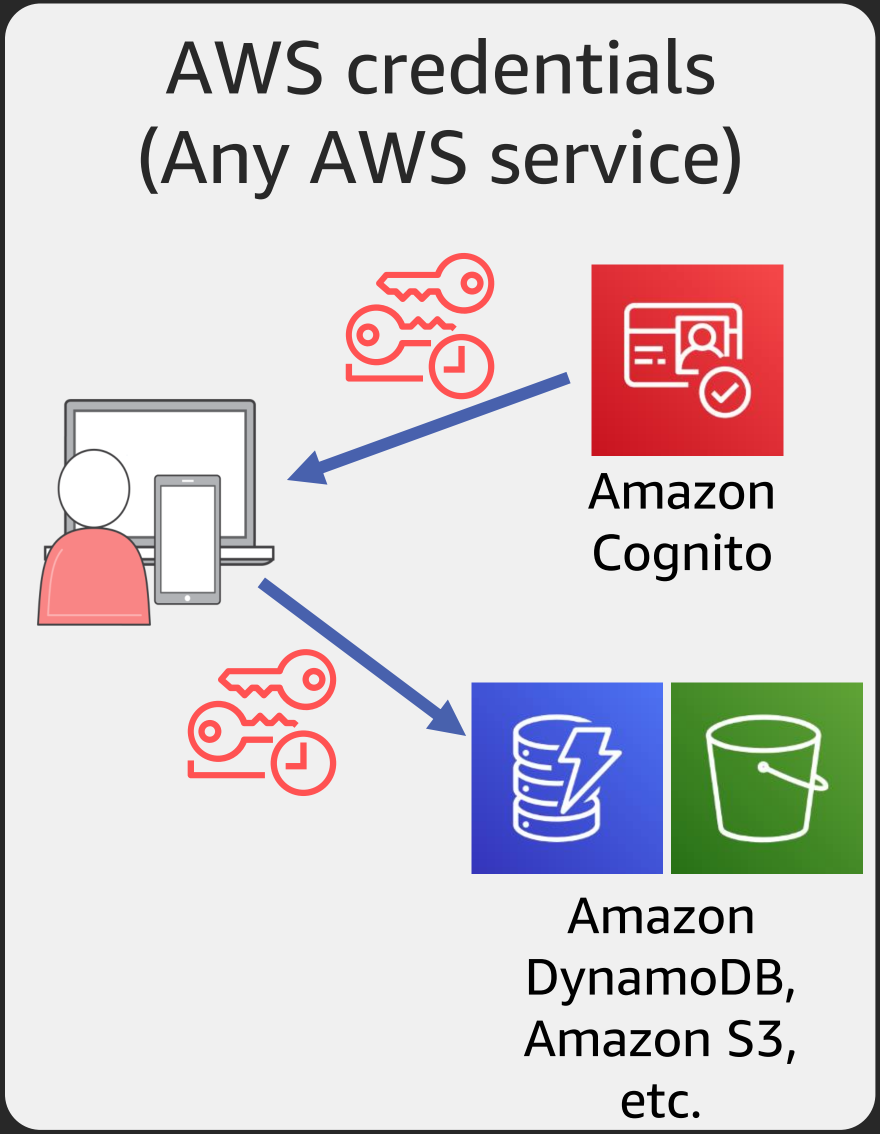
Amazon Cognito manages federation for you


IdP
P
P
IdP UI
4 User authenticated by IdP (SSO if active session)
Amazon Cognito
2 Determine IdP
Hosted UI
6 Create/Update profile
5 POST back with SAML assertion
Redirect to IdP 3
<SAML>
1 User directed to authenticate
Mobile or web app
OIDC token
7 Amazon Cognito tokens provided to app

Amazon Cognito integrations






AWS credentials (Any AWS service)
Amazon Cognito
Amazon DynamoDB, Amazon S3, etc.

Demo

Architecture overview

Where to handle application level access control?

Policy-based (e.g., IAM policies for API Gateway REST APIs)

- Centralized
- Declarative
- Role-based

Interceptor-based (e.g., API Gateway Lambda Custom Authorizer)

- Dynamic
- Attribute-based

In your application’s business logic code

- Access to domain objects (e.g., rule: user can trade options only if balance >= X)
- Ability to unit-test the authorization logic offline as part of CI/CD

“Authorization is a type of business logic . . .”

GraphQL Official Docs

https://graphql.org/learn/authorization/

"... some authorization logic is required within the business logic."

StackExchange.com

Top voted answer to "Where does authorization fit in a layered architecture?"

https://softwareengineering.stackexchange.com/a/294052/7685

“Authorization is business logic.”

Many people on the internet

General concept

IdP: LDAP Groups → SAML attributes

Amazon Cognito: SAML attributes → JIT user profile

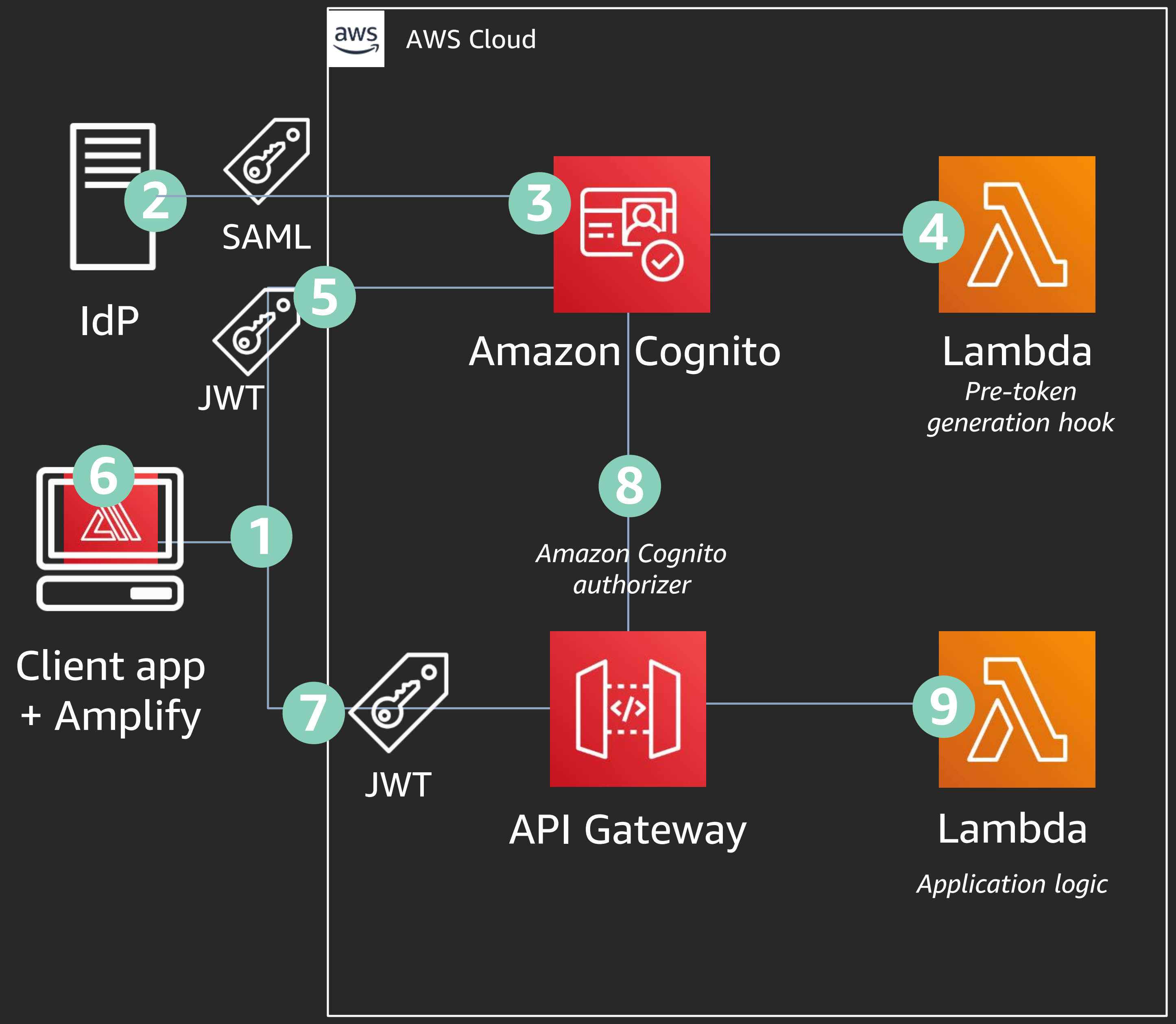
AWS Lambda Trigger*: JIT user profile → groups, roles → JWT

API Gateway: Validate JWT → app

App: Groups, roles from JWT → context aware authorization

*Optional

Architecture



1. AWS Amplify redirects if unauthenticated
2. IdP authenticates and sends SAML response
3. Amazon Cognito handles SAML response and maps SAML attributes to user profile
4. Lambda hook assigns users to roles/groups
5. Signed JWT tokens are returned to app
6. Amplify stores tokens and handles refreshes
7. Client app makes a call to a protected API
8. API Gateway (via Amazon Cognito user pools authorizer) validates the JWT token's signature and expiration and passes to Lambda, including token claims
9. Application code reads the group claims from the request and makes a decision if action is allowed

Deep dive into the code

Repository structure

- https://github.com/aws-samples/amazon-cognito-example-for-external-idp

- **/cdk**
  Infrastructure as code

- **/lambda/api**
  An express.js app using aws-serverless-express to run on Lambda

- **/lambda/pretokengeneration**
  An Amazon Cognito Lambda trigger to map IdP attribute to token claims (groups)

- **/ui-react**
  The frontend of our app uses Amplify to interact with Amazon Cognito

Assigning groups/roles

```
event.response = {
  claimsOverrideDetails: {
    claimsToSuppress,
    groupOverrideDetails: {
      // Will end up as a cognito:groups claim
      groupsToOverride: ldapGroupsArr,
    },
  },
};

return event;
```

See here for more details:
https://docs.aws.amazon.com/cognito/latest/developerguide/user-pool-lambda-pre-token-generation.html

Getting groups claim—backend

```
app.use(authorizationMiddleware({
  supportedGroups: [adminsGroupName, usersGroupName]
}));


app.get("/pets", async (req: Request, res: Response) => {

  if (req.groups.has(adminsGroupName)) {

    // if the user has the admin group, we return all pets
    res.json(await storageService.getAllPets());

  } else {

    // otherwise, just owned pets (middleware ensures user is in either of the 2 groups)
    res.json(await storageService.getAllPetsByOwner(req.username));

  }
});
```

Getting groups claim—frontend

```
import {CognitoUser} from "@aws-amplify/auth";

export class User {

  constructor(private cognitoUser: CognitoUser) {
  }

  getGroups(): string[] {
    return this.getClaims()["cognito:groups"] || [];
  }

  getClaims(): { [id: string]: any; } {
    return this.cognitoUser && this.cognitoUser.getSignInUserSession()
      && this.cognitoUser.getSignInUserSession().isValid()
      && this.cognitoUser.getSignInUserSession().getIdToken().decodePayload() || {};
  };
}


groups = user.getGroups();
```

Other resources

Amazon Cognito documentation—adding identity providers
https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-identity-federation.html

AWS serverless express

https://github.com/awslabs/aws-serverless-express/

Amplify

https://aws.amazon.com/amplify/

AWS CDK

https://docs.aws.amazon.com/cdk/

Learn security with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate cloud security skills

- 30+ free digital courses cover topics related to cloud security, including Introduction to Amazon GuardDuty and Deep Dive on Container Security

- Classroom offerings, like AWS Security Engineering on AWS, feature AWS expert instructors and hands-on activities

- Validate expertise with the **AWS Certified Security - Specialty** exam

Visit aws.amazon.com/training/paths-specialty/

Thank you!

Eran Medan

eranmeda@amazon.com

Please complete the session survey in the mobile app.

AWS re:Invent

aws