
AWS
re:Invent
DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

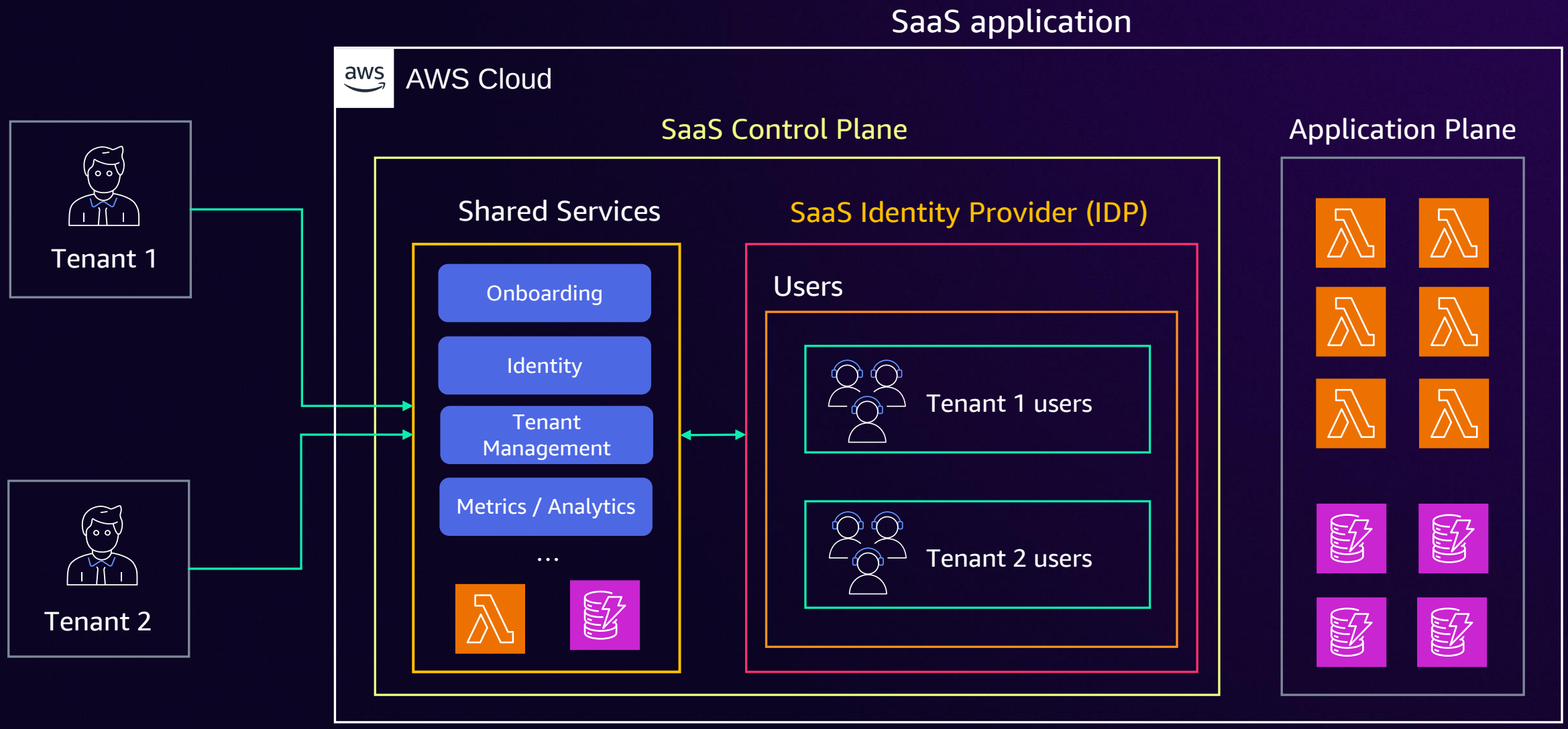
SAS307

Supporting federated identity in multi-tenant environments

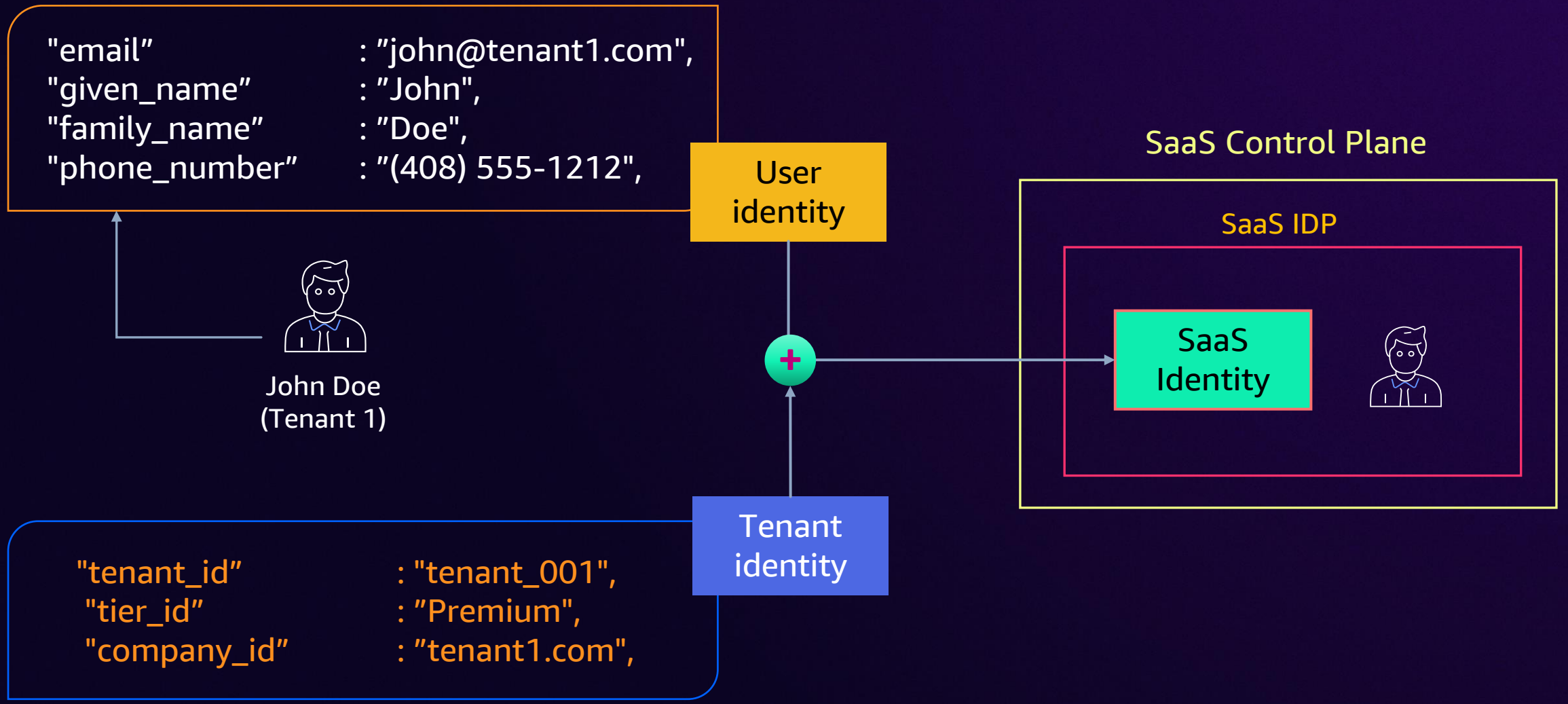
Toby Buckley
(he/him)
Sr. Partner Solution Architect
AWS SaaS Factory

Dhammika Sriyananda
(he/him)
Sr. Partner Solution Architect
AWS SaaS Factory

Managing identities in a SaaS



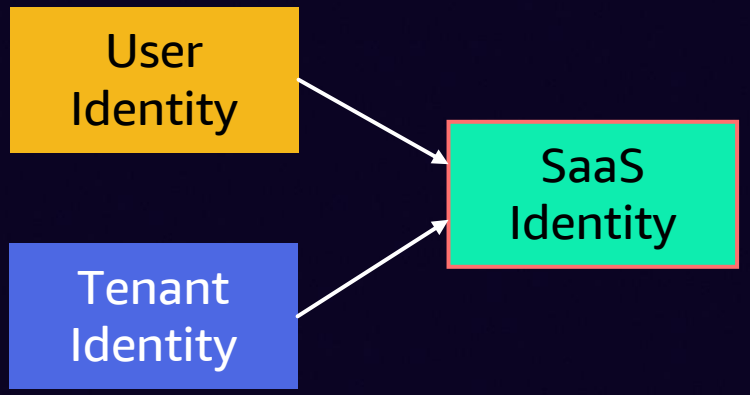
Representing SaaS identity



A first-class SaaS token to represent SaaS identity

Identity Tokens

- Fully managed by the SaaS IdP
- Security in-built
- Embedded the SaaS Identity



JSON Web Token (JWT)

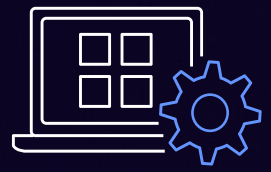
```
{
  "sub"              : "95daf444-32gs",
  "email"            : "john@tenant1.com",
  "email_verified"   : true,
  "given_name"       : "John",
  "family_name"      : "Doe",
  "phone_number"     : "(408) 555-1212",
  . . .

  "tenant_id"        : "tenant_001",
  "tier_id"          : "Premium",
  "company_id"       : "tenant1.com",
  "status"           : "Active",
    . . .
}
```

User Details

Tenant Details

User auth flow with Amazon Cognito IDP

SaaS IDP



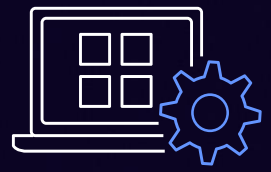
SaaS App

Amazon Cognito

John Doe
(Tenant 1)

User auth flow with Amazon Cognito IDP

SaaS IDP

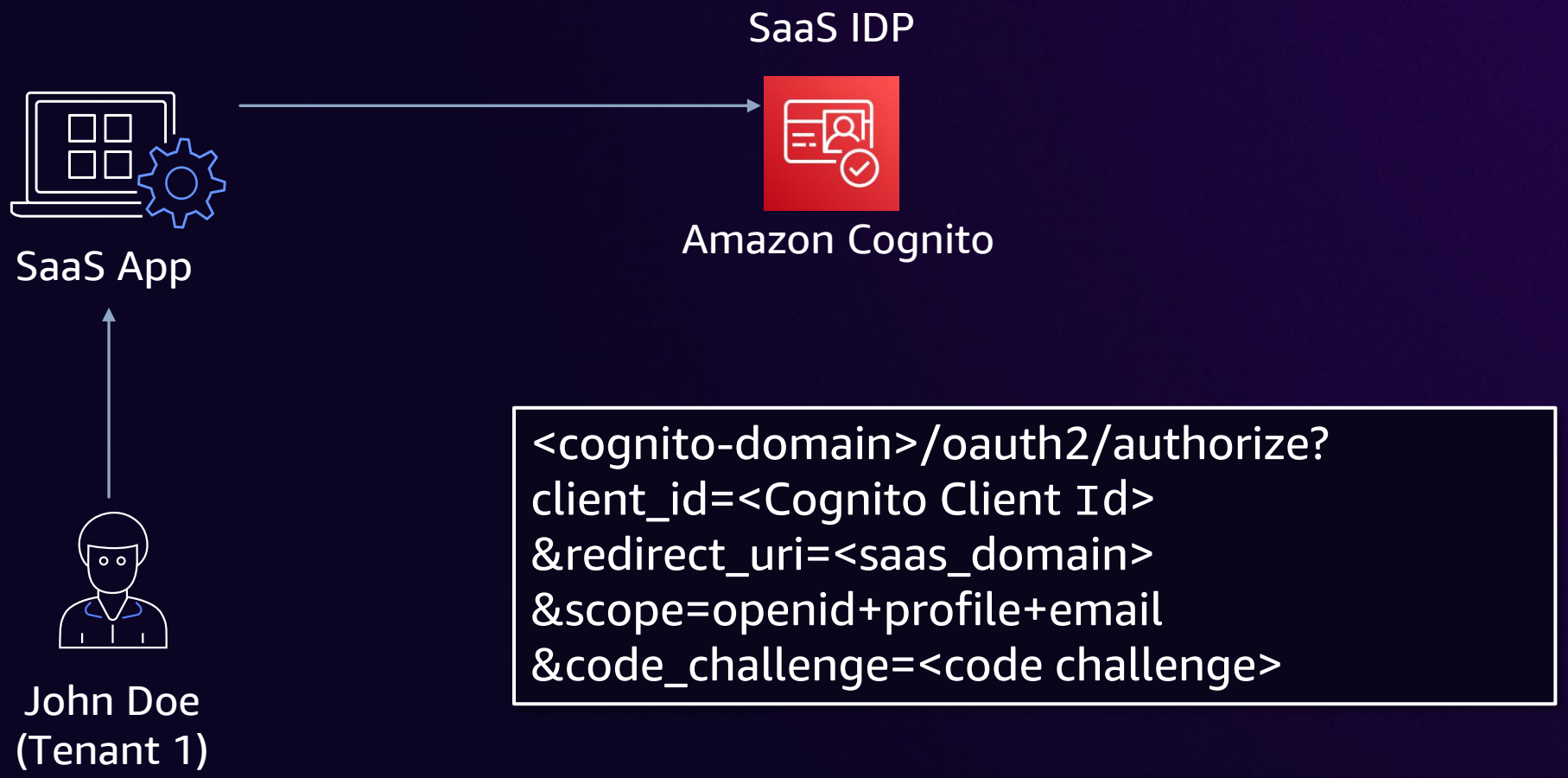


SaaS App

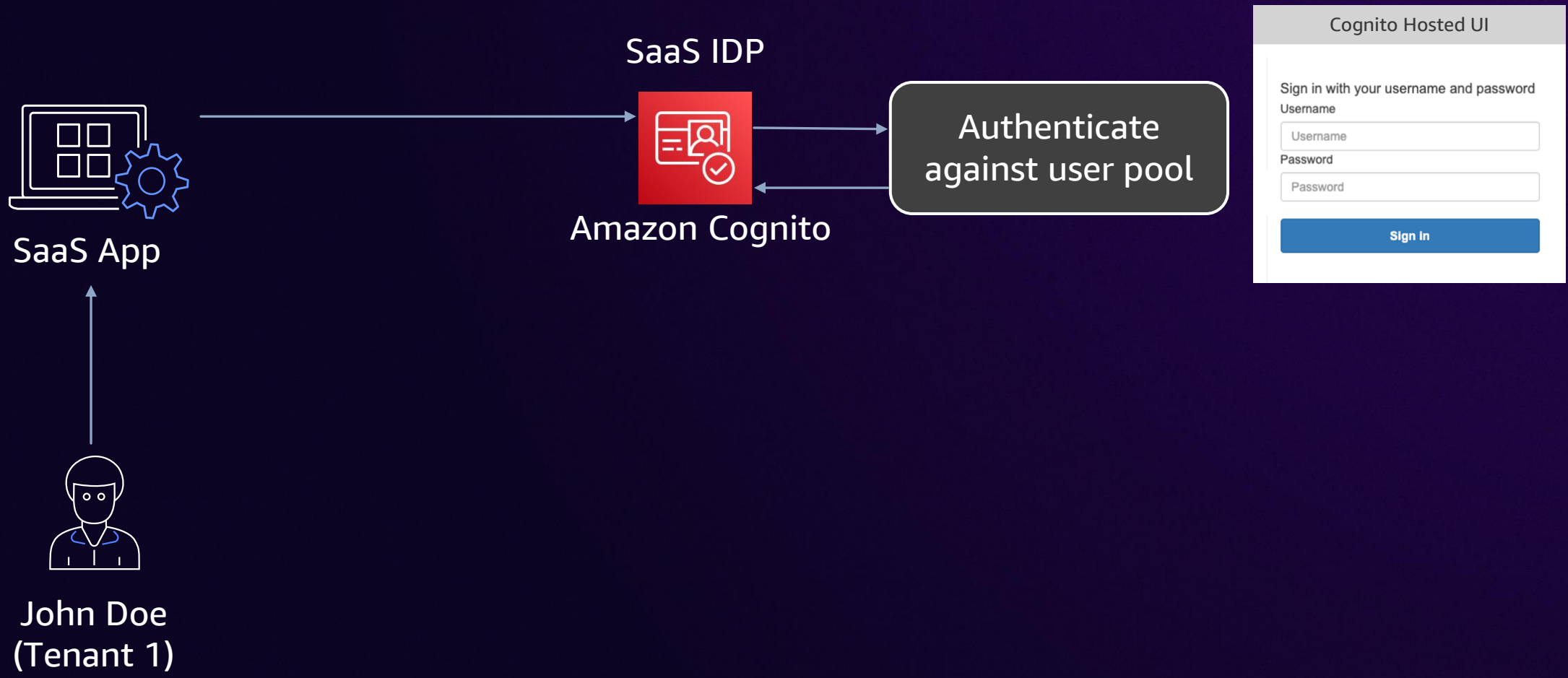
Amazon Cognito

John Doe
(Tenant 1)

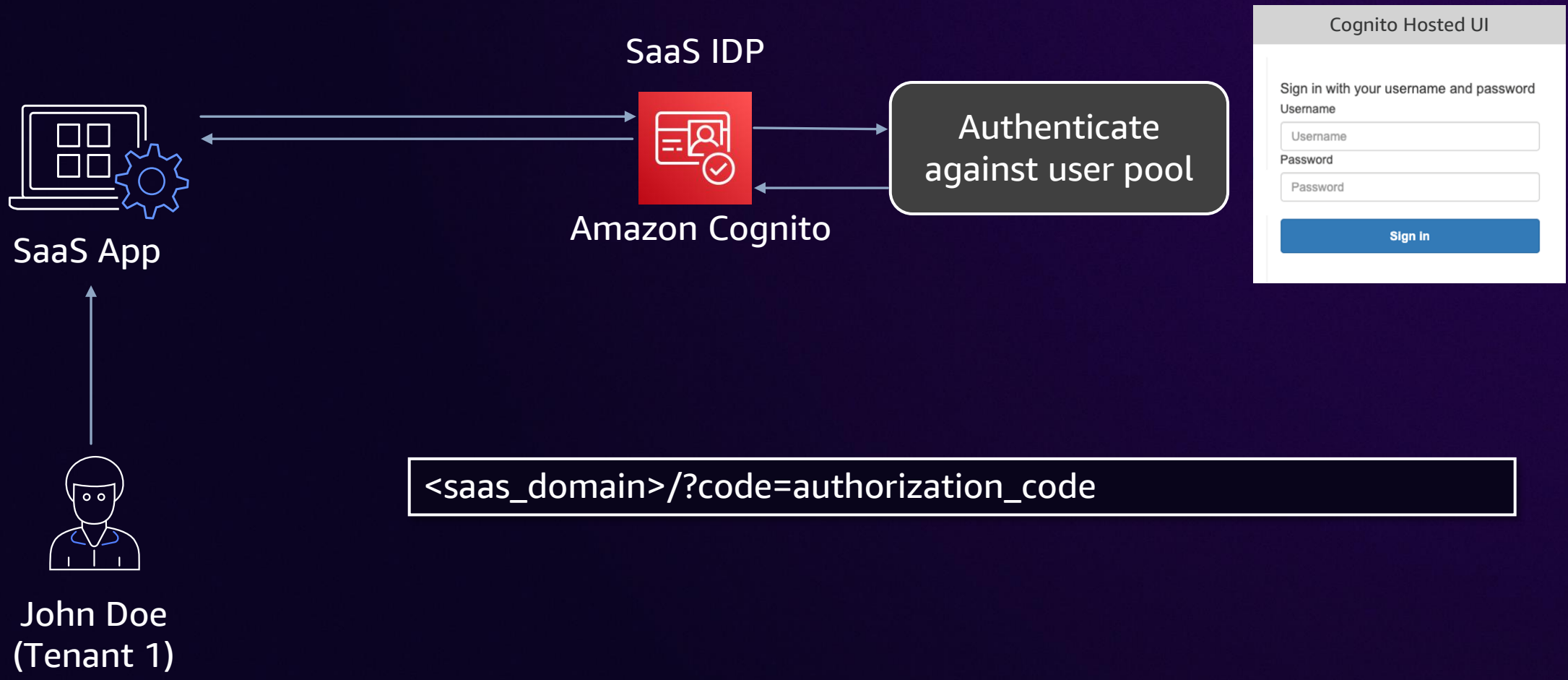
User auth flow with Amazon Cognito IDP



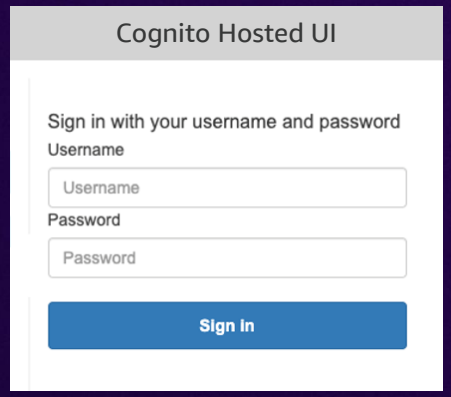
User auth flow with Amazon Cognito IDP



User auth flow with Amazon Cognito IDP



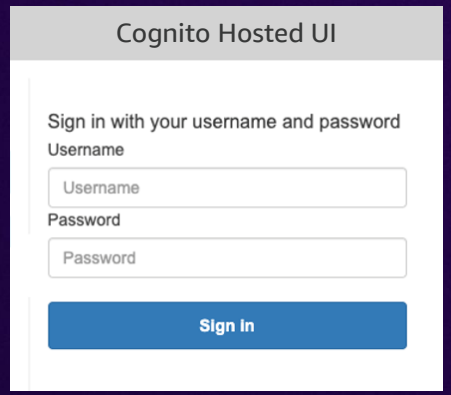
User auth flow with Amazon Cognito IDP



John Doe
(Tenant 1)

```
POST <cognito-domain>/oauth2/token
grant_type=authorization_code
&client_id=<Cognito Client Id>
&code_verifier=<code verifier>
&redirect_uri=<saas_domain>
```

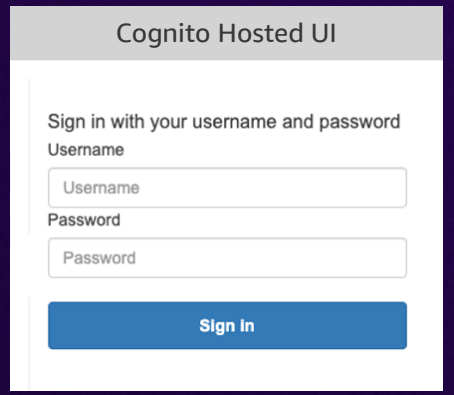
User auth flow with Amazon Cognito IDP



John Doe
(Tenant 1)

```
{
   "id_token": "<id_token>",
   "access_token": "<access_token>",
   "refresh_token": "<refresh_token>",
   "expires_in": 3600,
   "token_type": "Bearer"
}
```

User auth flow with Amazon Cognito IDP

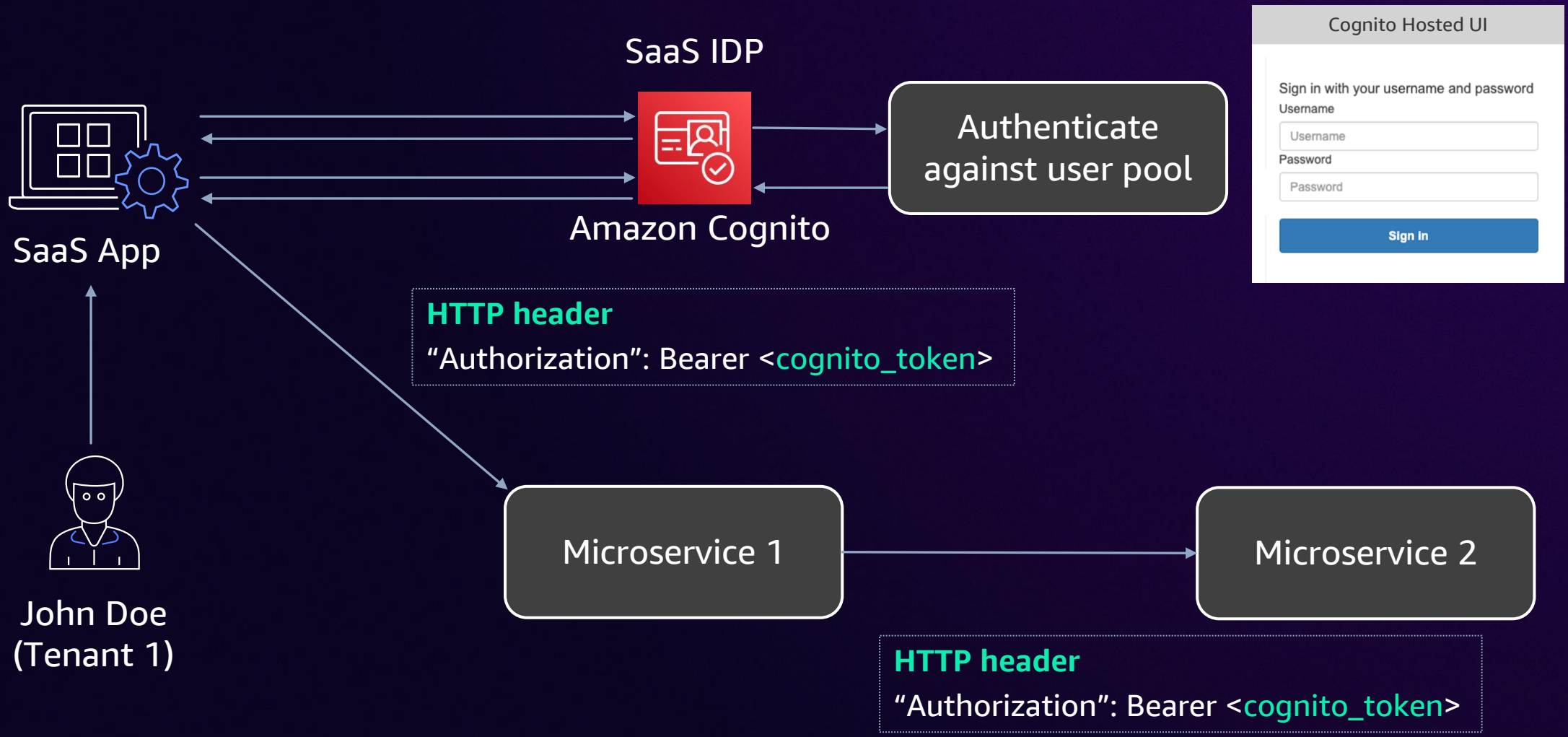


John Doe
(Tenant 1)

Cognito JSON Web Token (JWT)

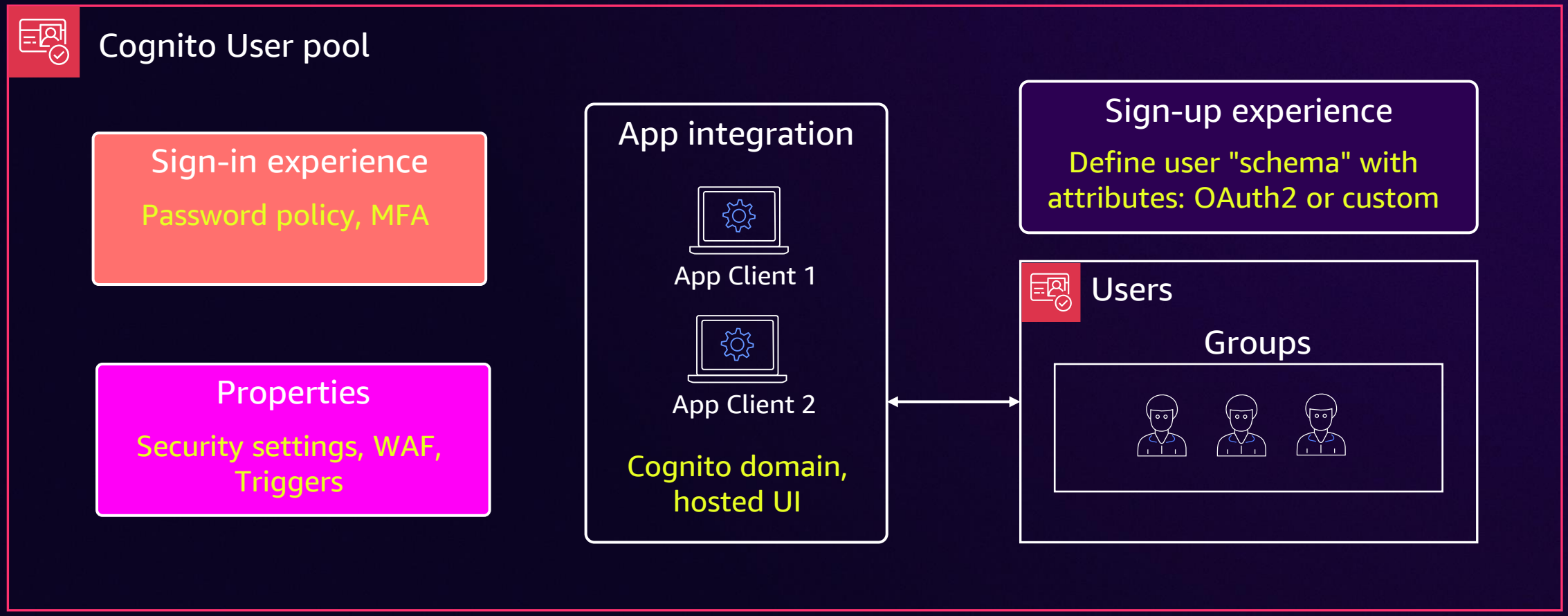
```
{
  "sub"                    : "95daf444-32gs",
  "email"                  : "john@tenant1.com",
  "email_verified"         : true,
  "given_name"             : "John",
  "family_name"            : "Doe",
  "phone_number"           : "(408) 555-1212",
  "custom:tenant_id"       : "tenant_001",
  "custom:tier_id"         : "Premium",
  "custom:company_id"      : "tenant1.com",
  "custom:status"          : "Active",
}
```

User auth flow with Amazon Cognito IDP

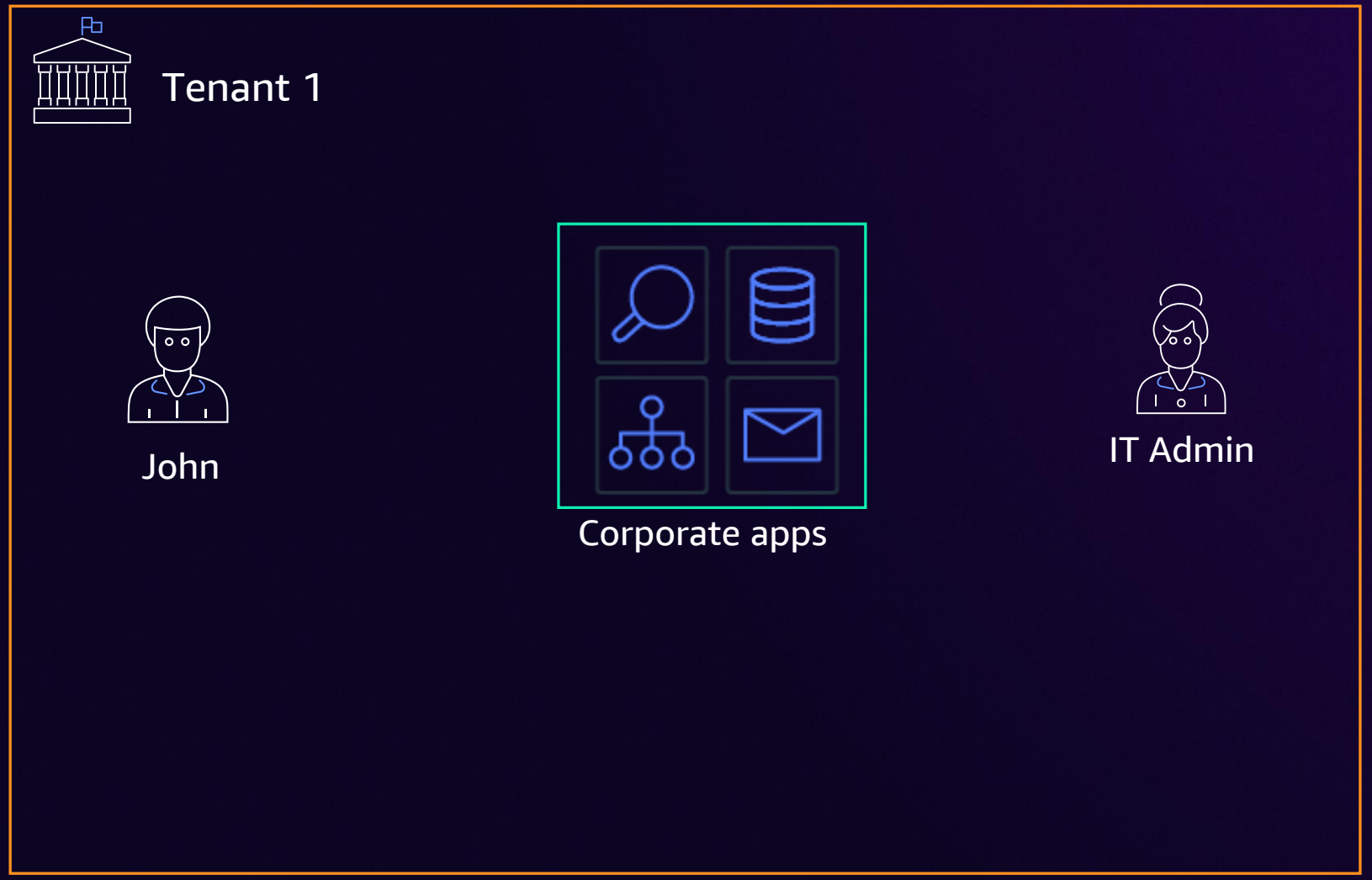


Amazon Cognito as the SaaS IDP

Cognito user pool elements that help to represent SaaS identity



Tenants demand a “single source of identity”



SaaS Application

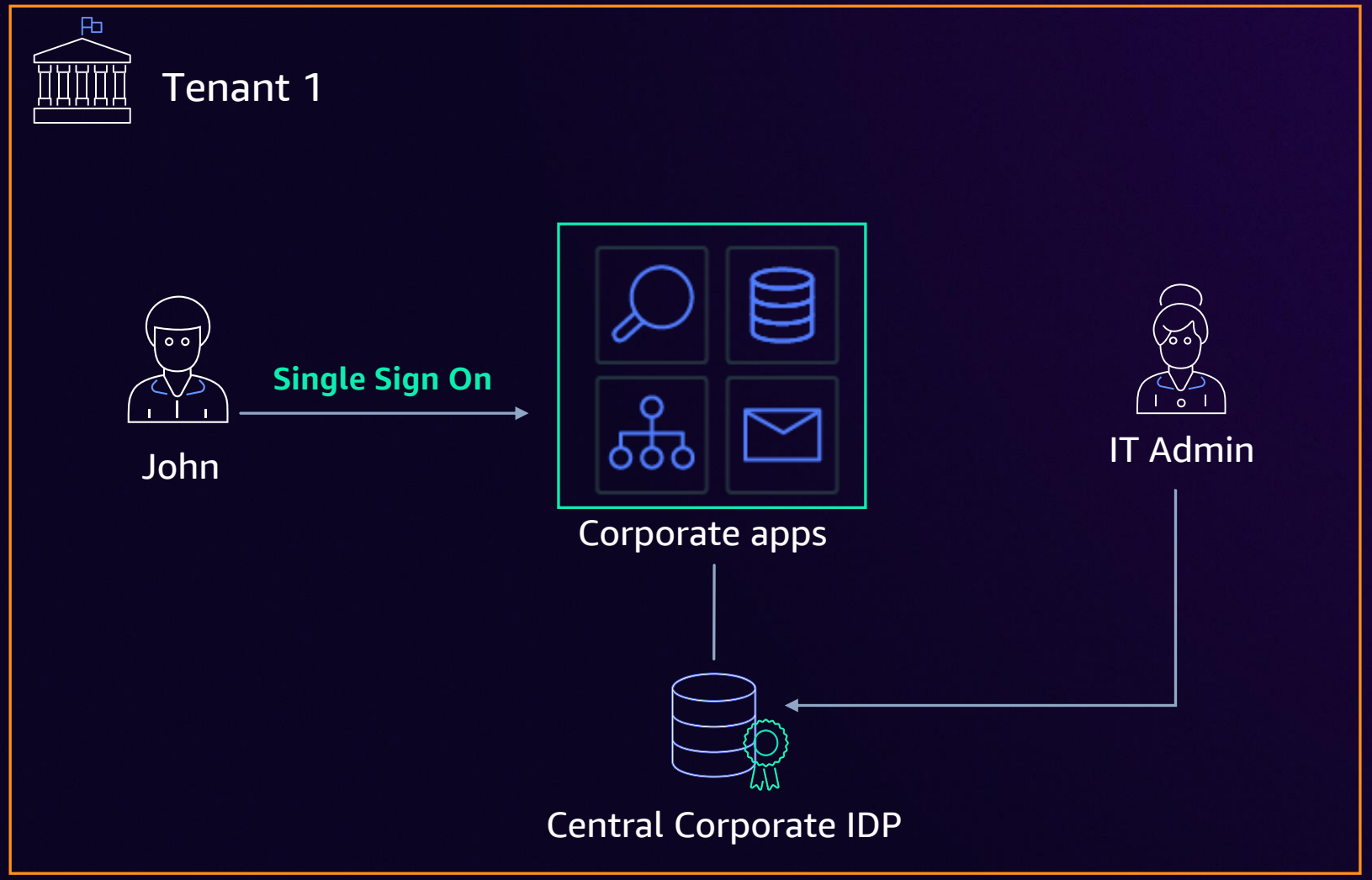
Tenant Id	Name
1	Nikki
1	John
1	Jorge

Tenants demand a “single source of identity”

SaaS Application

Tenant Id	Name
1	Nikki
1	John
1	Jorge

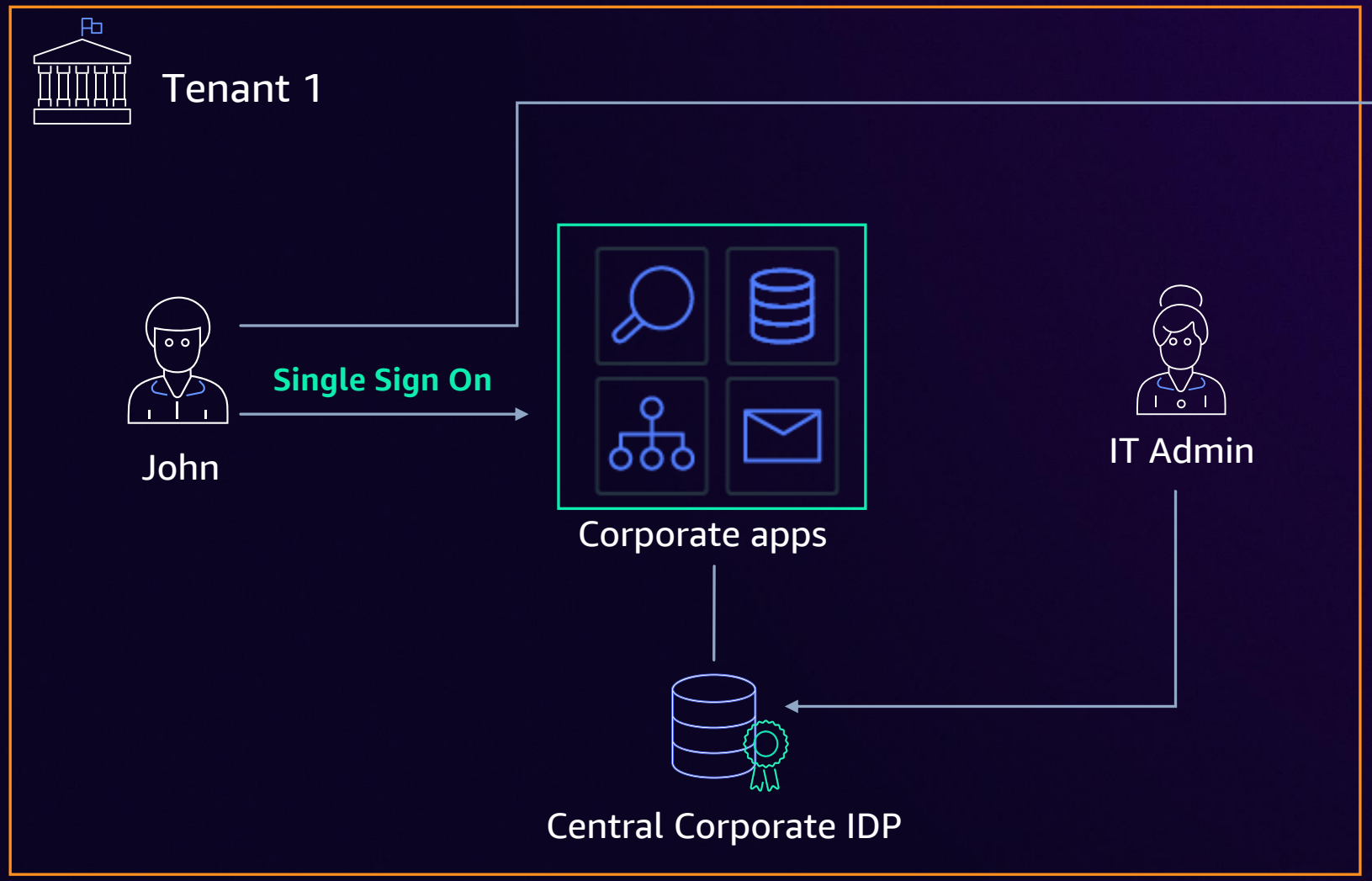
Tenants demand a “single source of identity”



SaaS Application

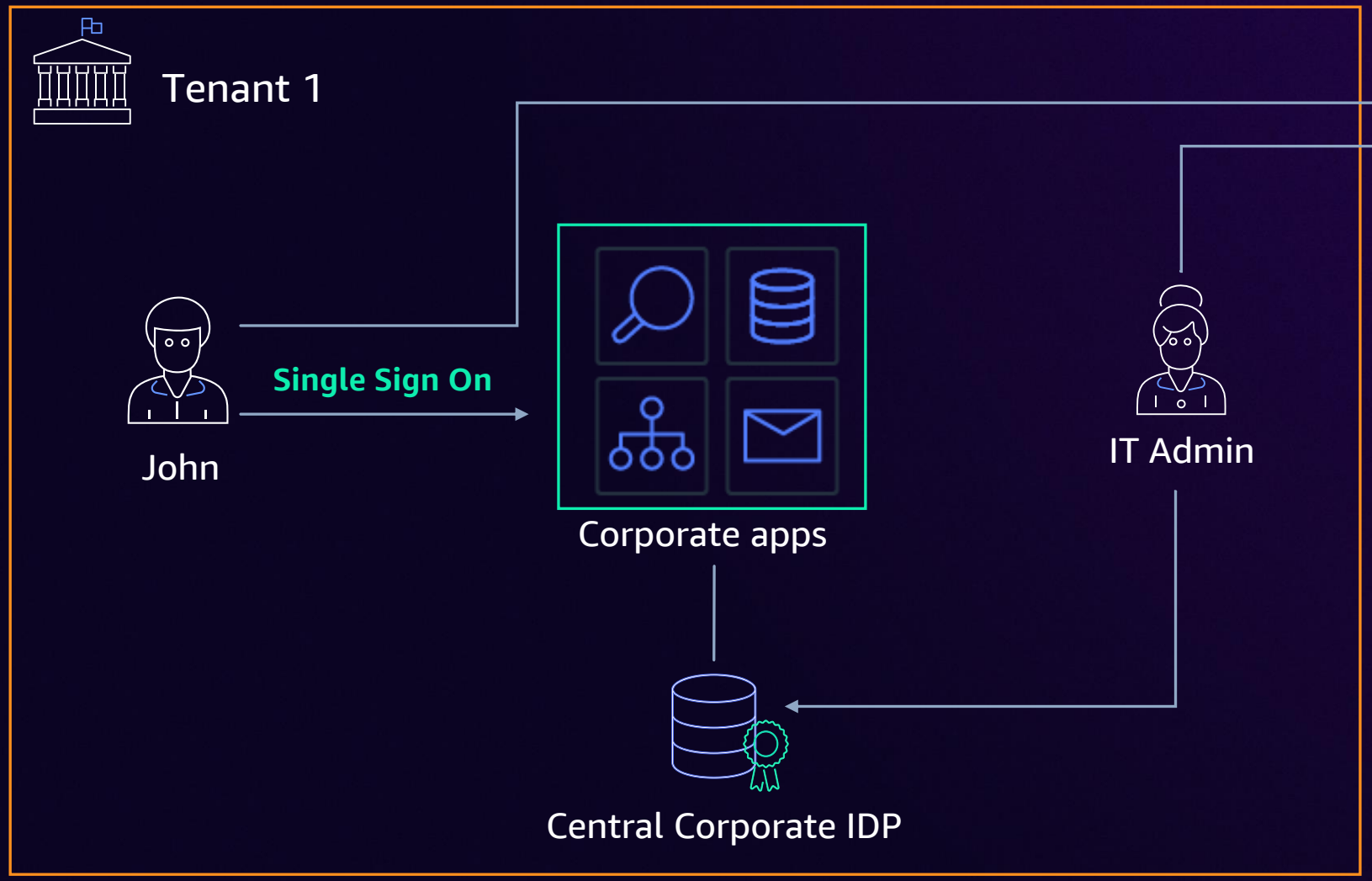
Tenant Id	Name
1	Nikki
1	John
1	Jorge

Tenants demand a “single source of identity”



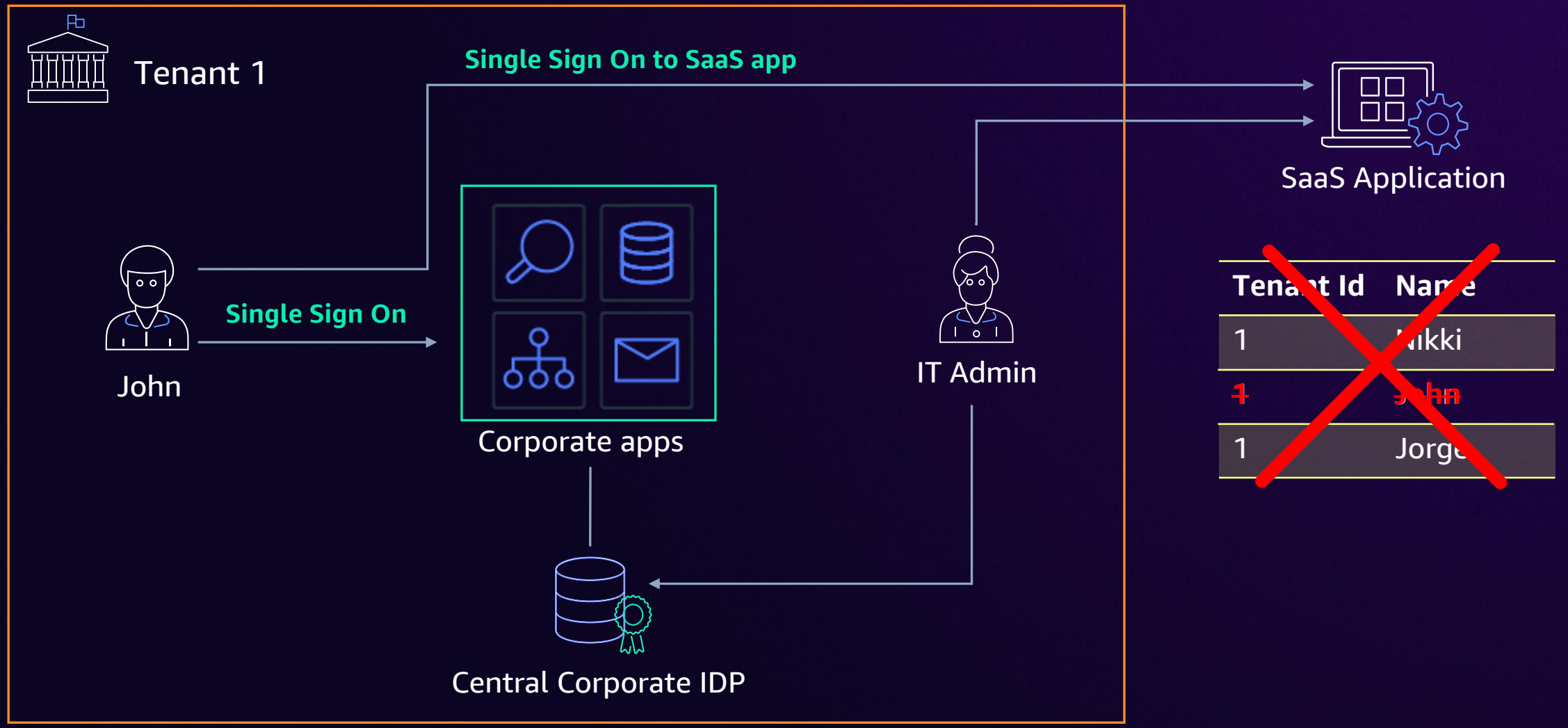
Tenant Id	Name
1	Nikki
1	John
1	Jorge

Tenants demand a “single source of identity”

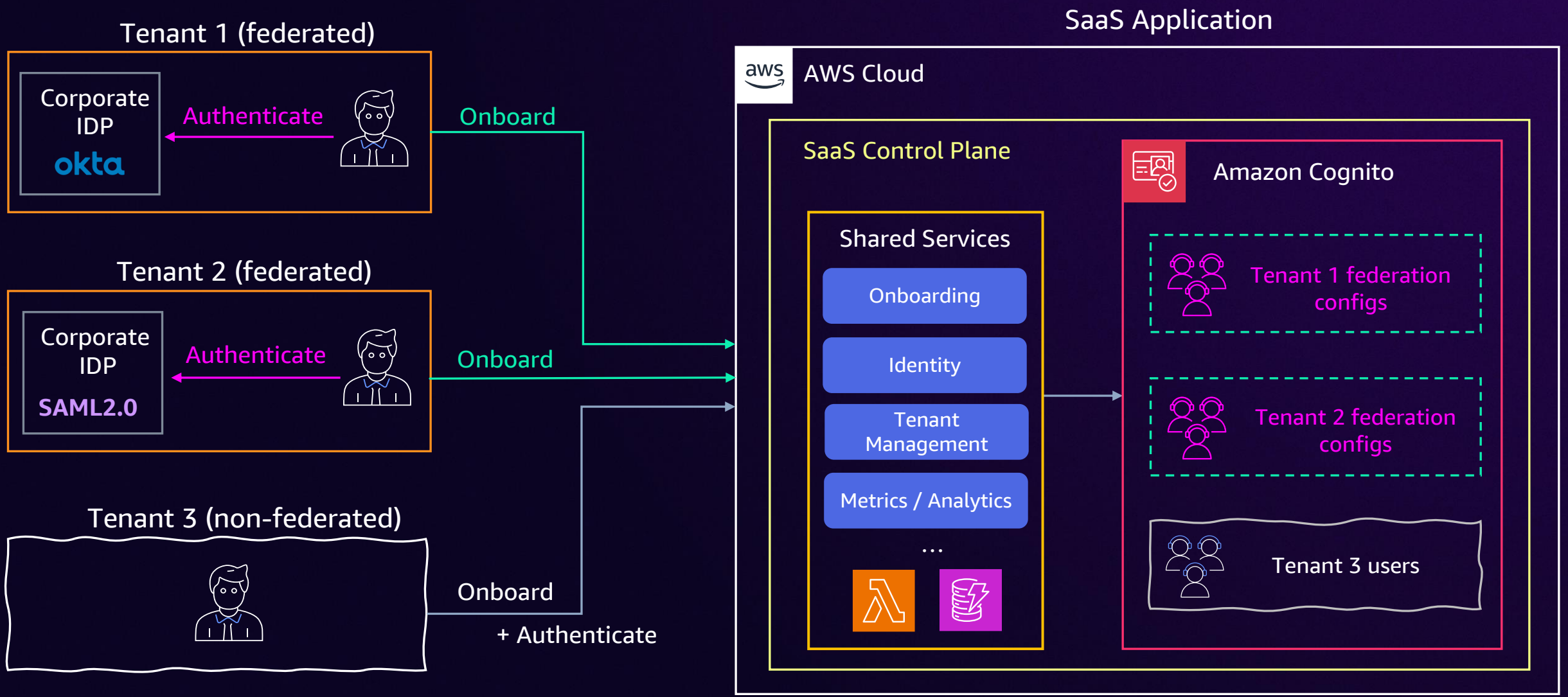


Tenant Id	Name
1	Nikki
~~1~~	~~John~~
1	Jorge

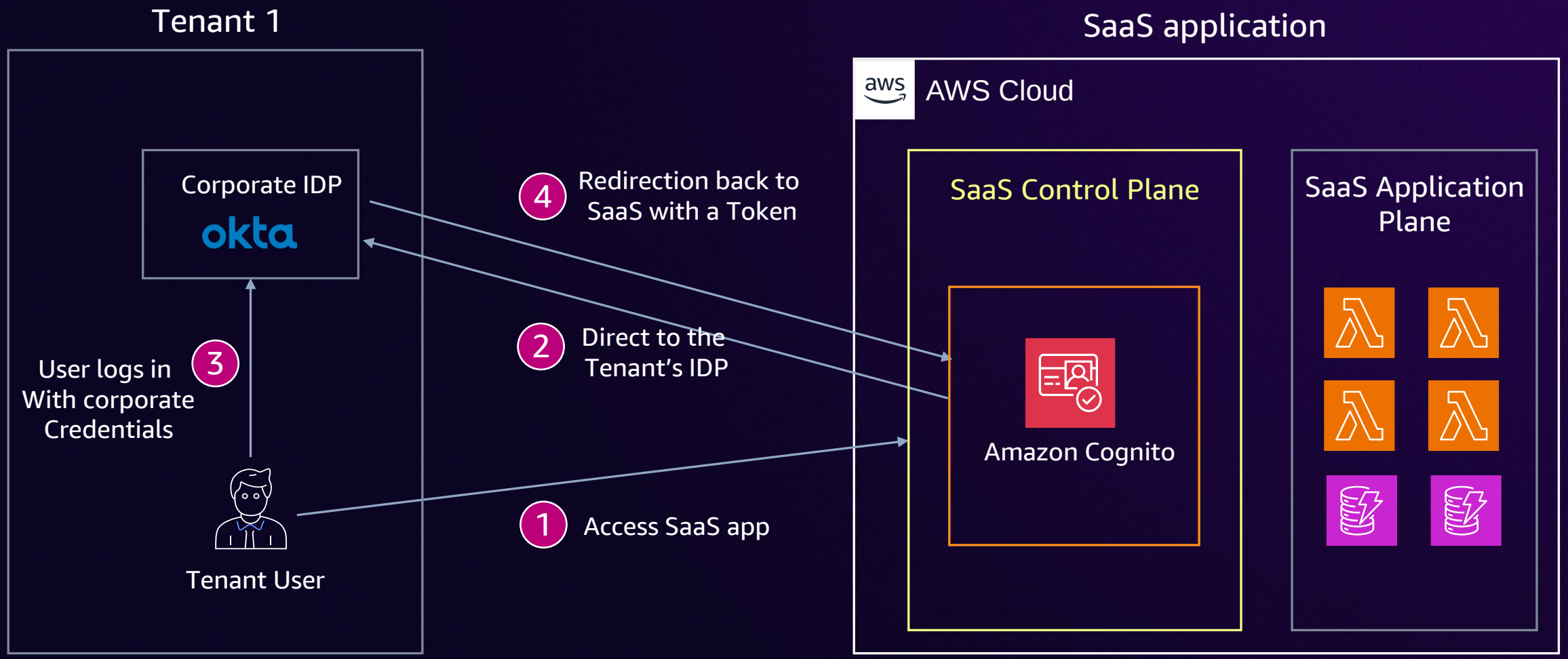
Tenants demand a “single source of identity”



SaaS need federated identity support

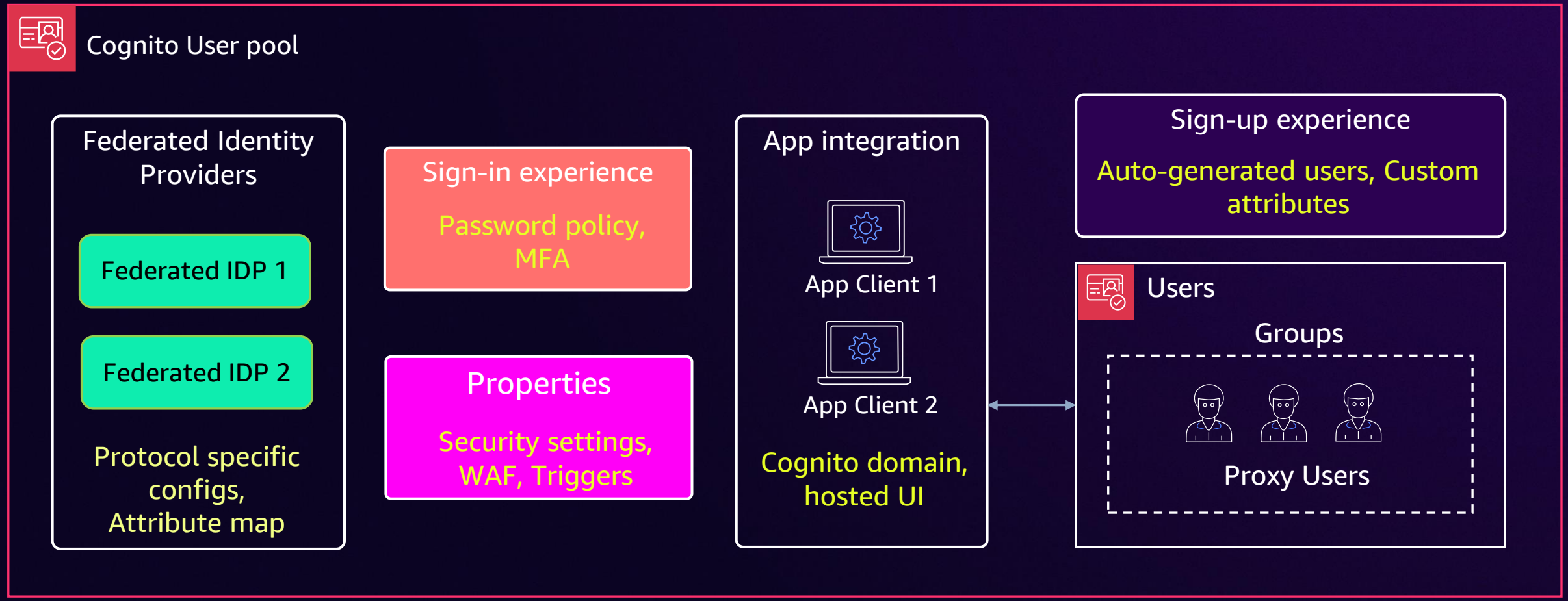


SaaS federated identity flow



Amazon Cognito as federated SaaS IDP

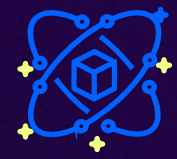
Cognito user pool elements that help to implement federated SaaS Identity



Federated SaaS identity – Challenges

Support both federation with standard identity flow in the SaaS

Need a generic way to support IDP integrations



Still need to support SaaS Identity = (user + tenant) identity

Handle compliance, cost and IDP quotas

Automate workflows for seamless federated experience

Identity federation protocols

OAuth/OIDC

An identity layer on top of OAuth 2.0. Allows third-party apps to verify user identities

SAML 2.0

Simplifies the user authentication using XML based communication

Social Providers

Sign-in through social identity providers such as Google, Amazon, Apple

Tenant onboarding with federated identity setup

Tenant onboarding challenge

Callback URL (OIDC) **or SSO URL** (SAML)

For Tenant1 IDP to POST the federated response back to SaaS

Tenant 1

Admin

Create an “Integration App”

Tenant1 IDP

“Integration App” Configs

OIDC

```
{
  “AppClientId”
  ”ClientSecret
  ”IssuerUrl”
}
```

OR

SAML

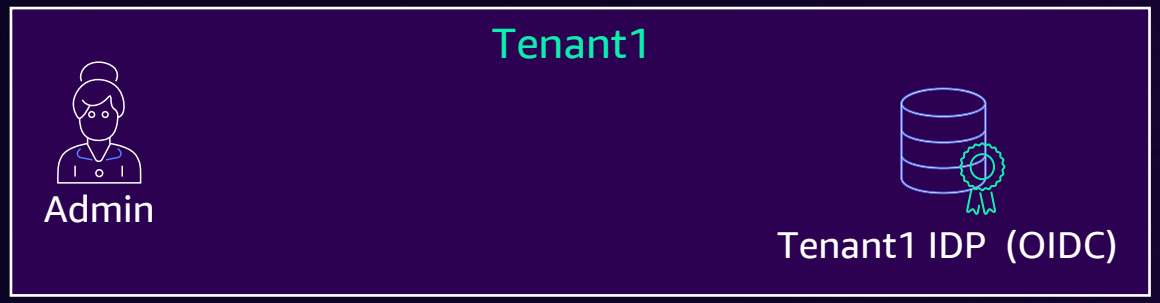
```
{
  “MetadataURL”
}
```

SaaS Application

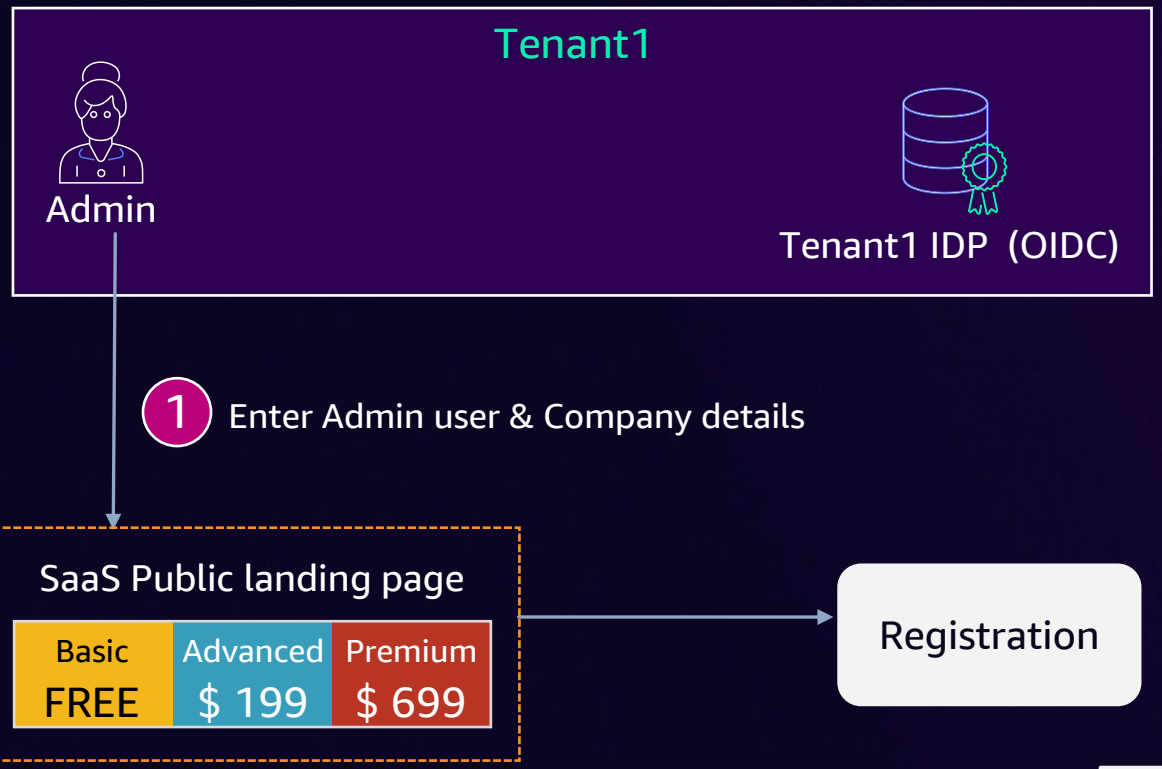
SaaS IDP

Amazon Cognito

Flow 1 : Federated tenant onboarding

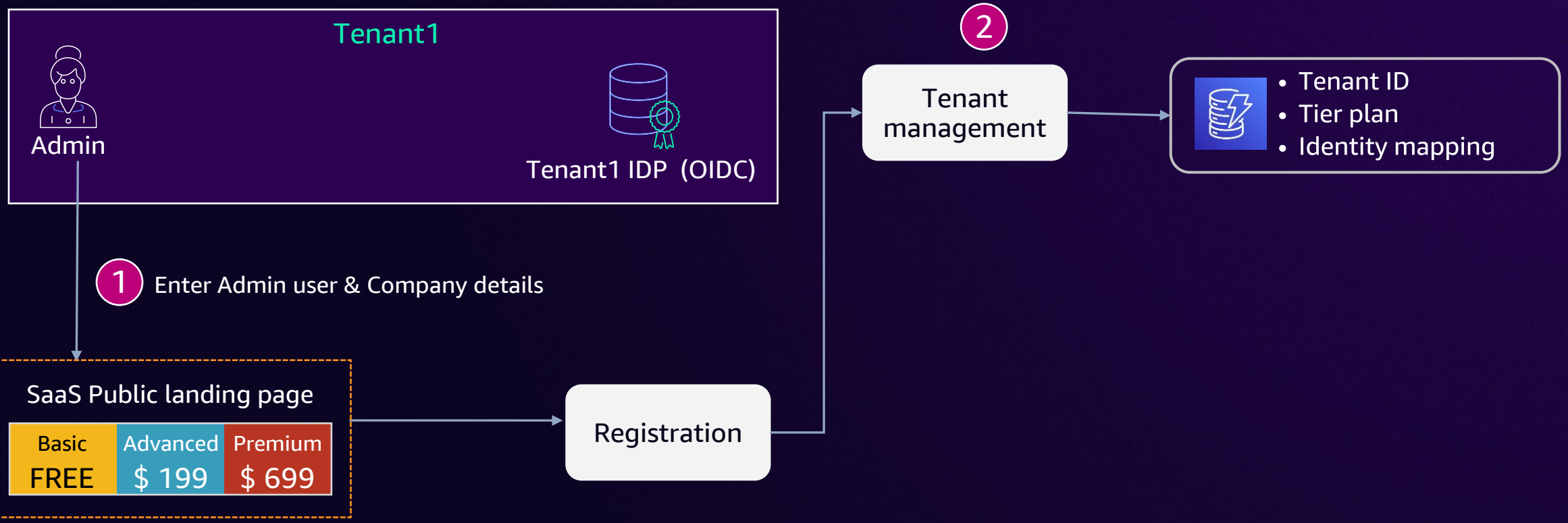


Flow 1 : Federated tenant onboarding

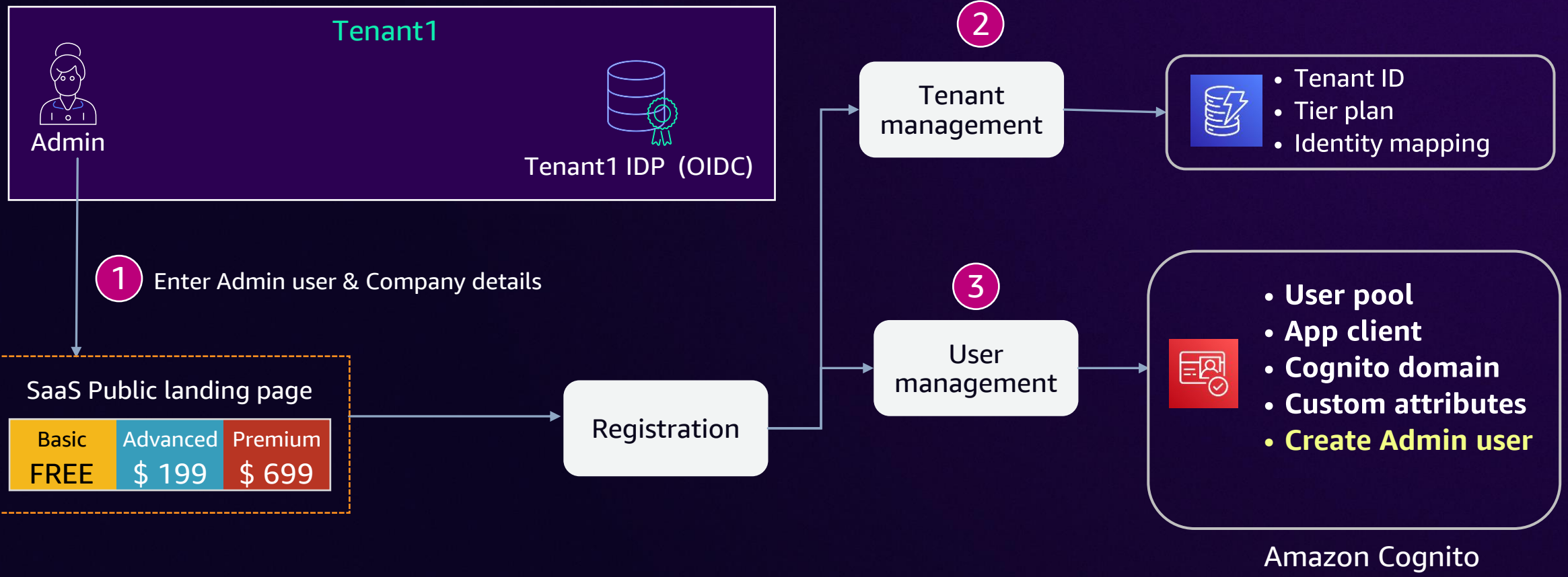



```
{
  ”adminName”: ” Jane Doe",
  ”adminEmail”: ”jane@tenant1.com",
  ”tier”: ”Premium",
  ”companyName”:”Tenant1",
  "companyURL”:” https://tenant1.com”
}
```

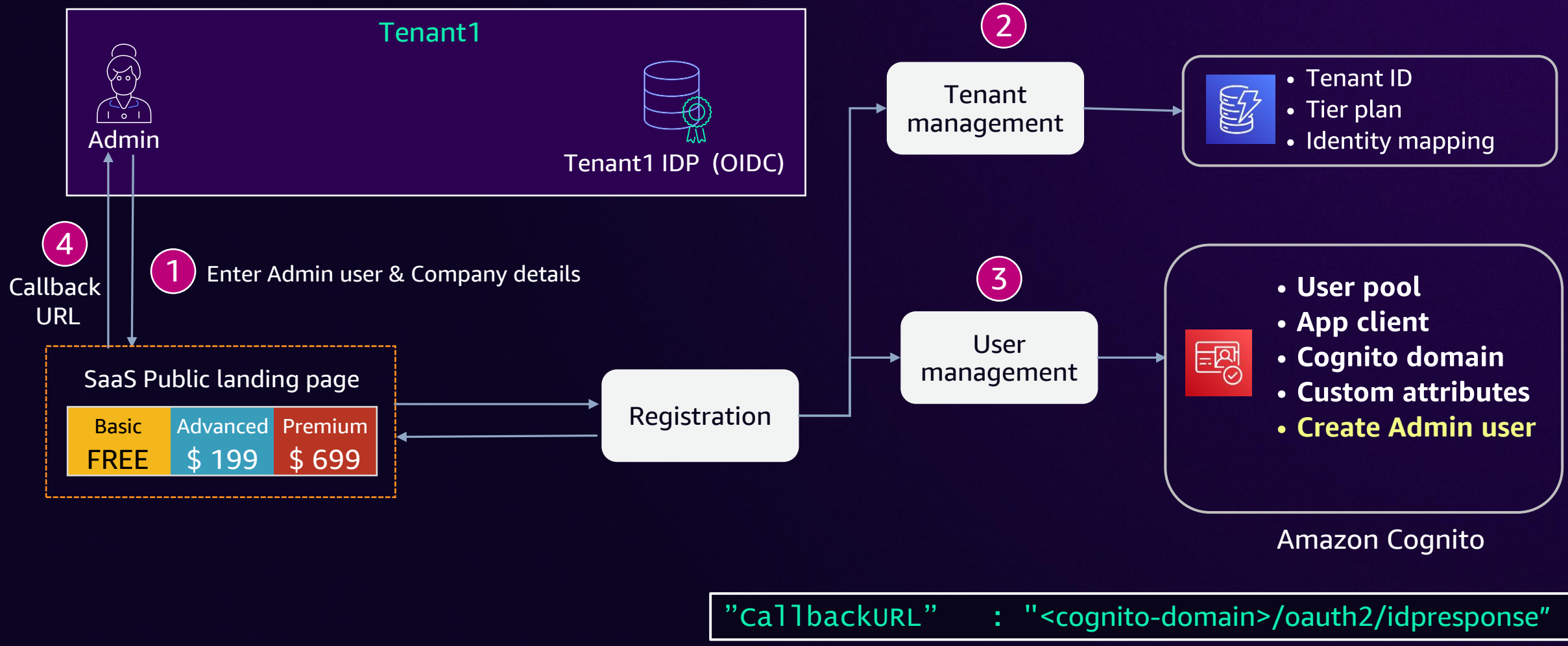
Flow 1 : Federated tenant onboarding



Flow 1 : Federated tenant onboarding

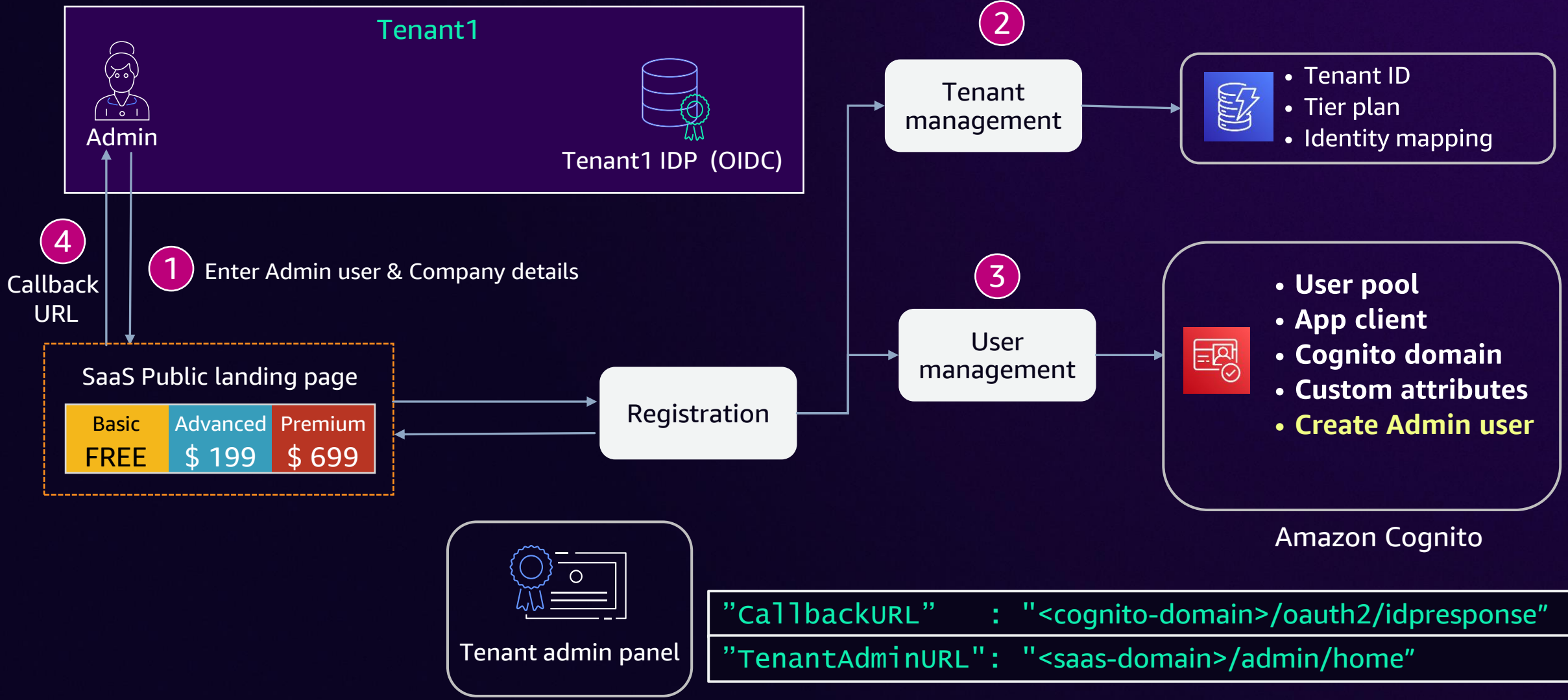


Flow 1 : Federated tenant onboarding

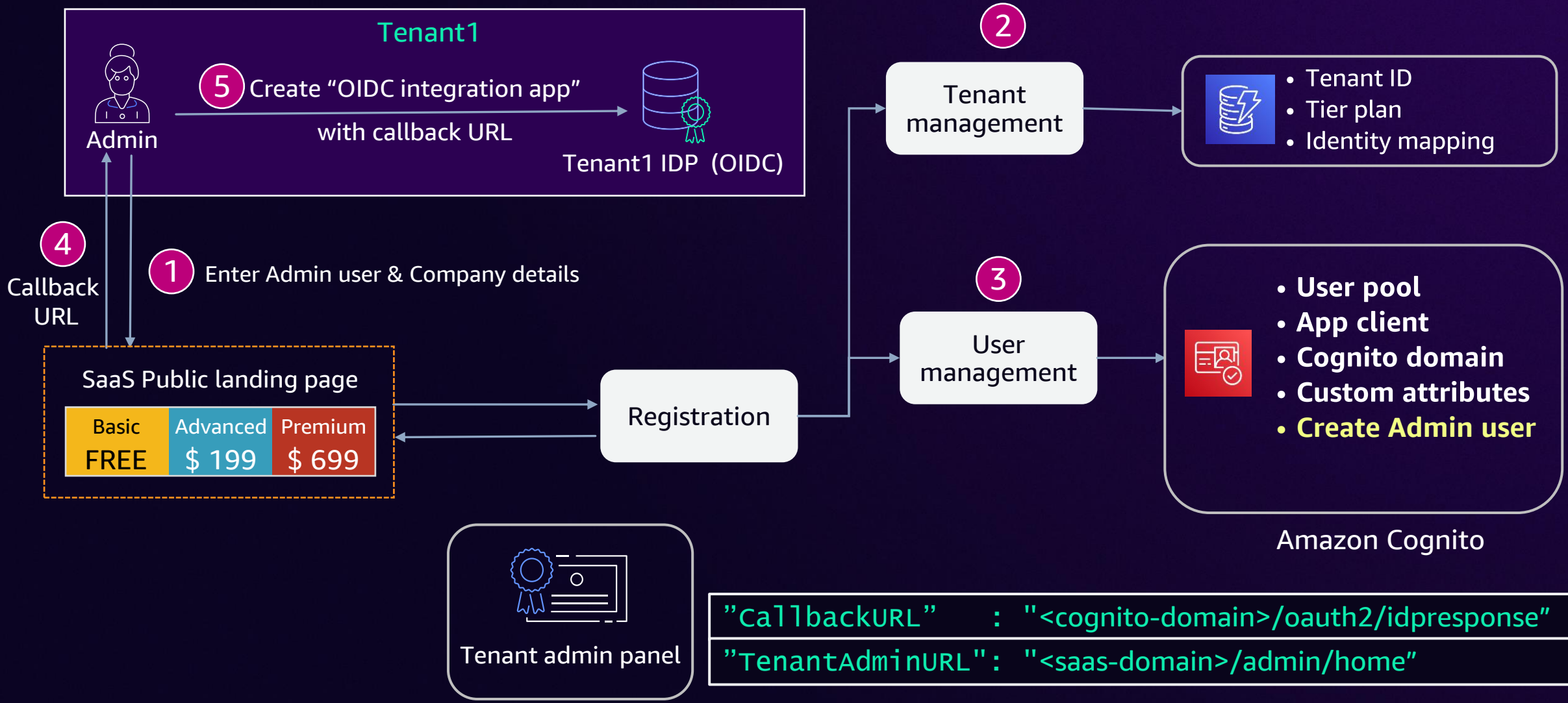



```
"callbackURL"  :  "<cognito-domain>/oauth2/idpresponse"
```

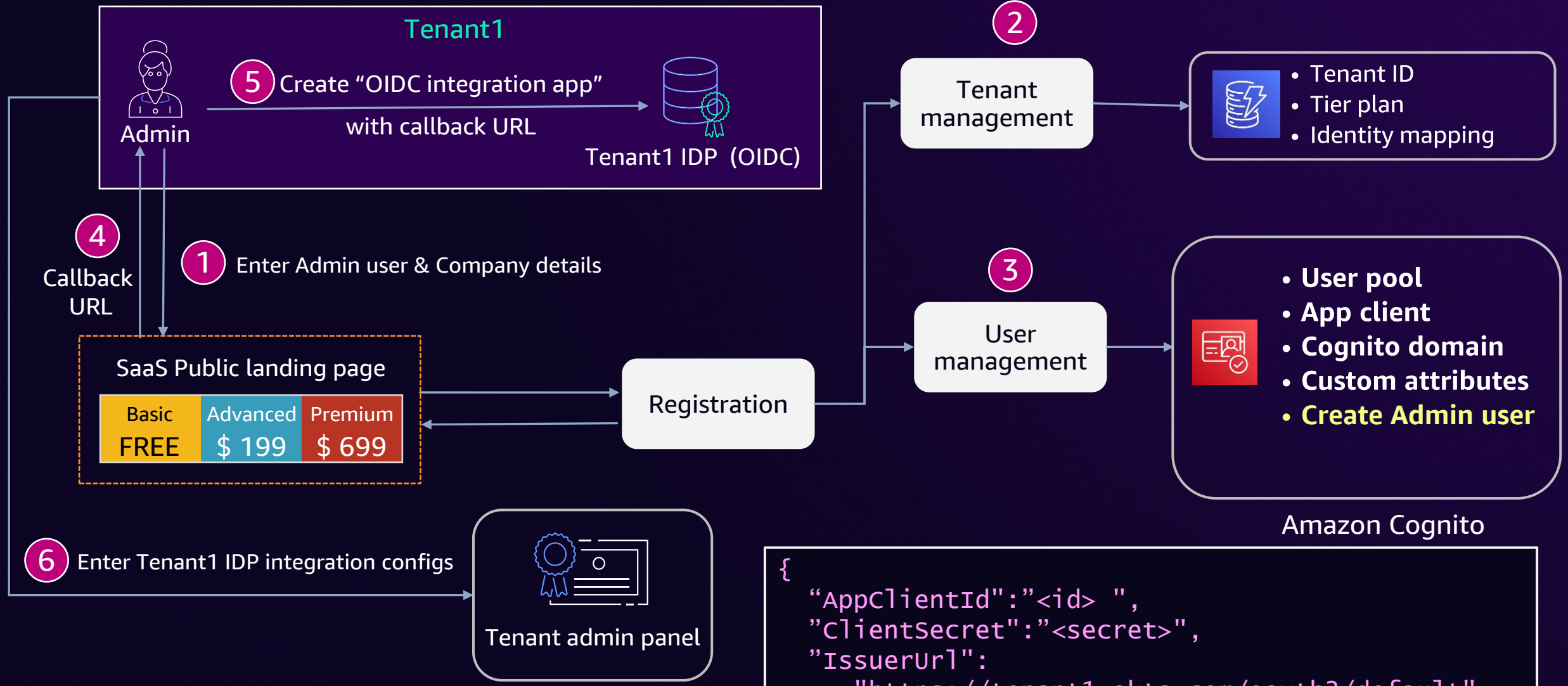
Flow 1 : Federated tenant onboarding



Flow 1 : Federated tenant onboarding

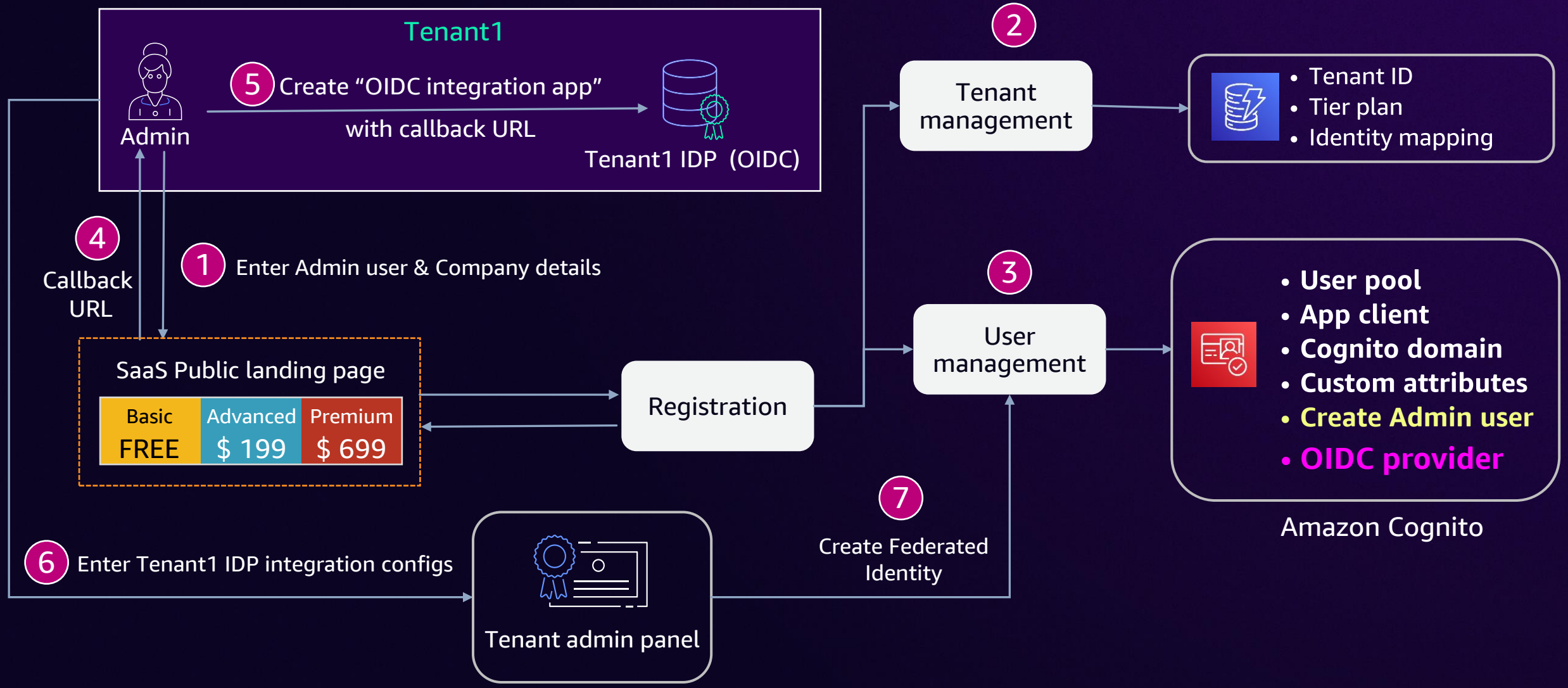


Flow 1 : Federated tenant onboarding

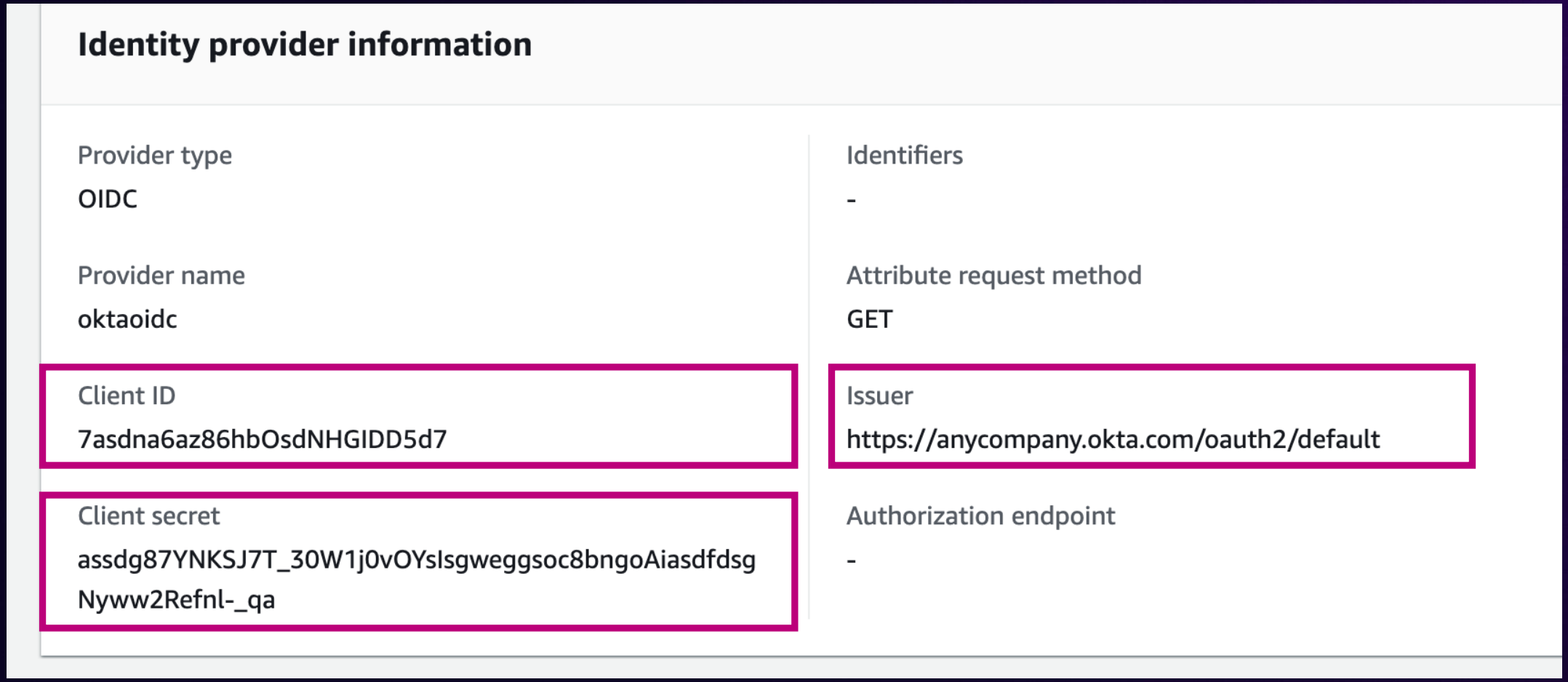



```
{
  “AppClientId":”<id> ",
  ”ClientSecret":”<secret>",
  ”IssuerUrl":
     "https://tenant1.okta.com/oauth2/default"
}
```

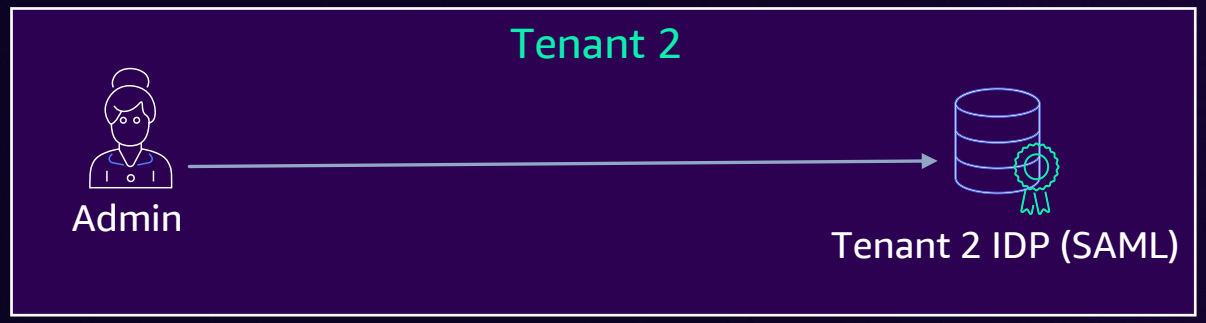
Flow 1 : Federated tenant onboarding



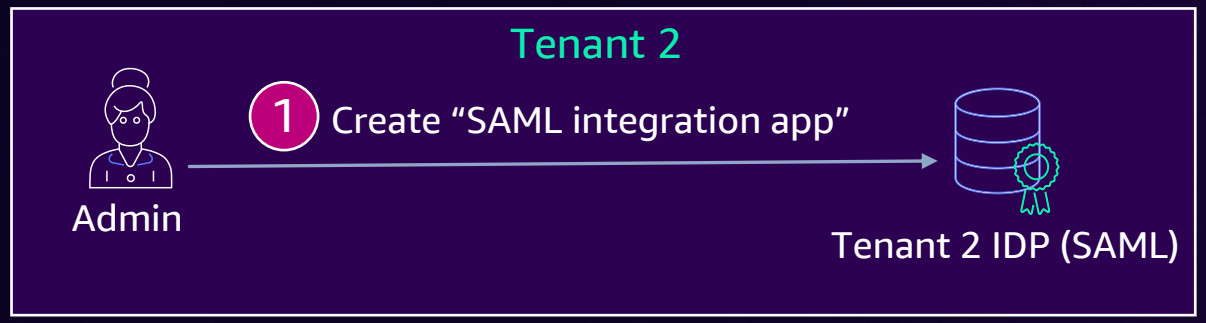
Amazon Cognito – OIDC provider



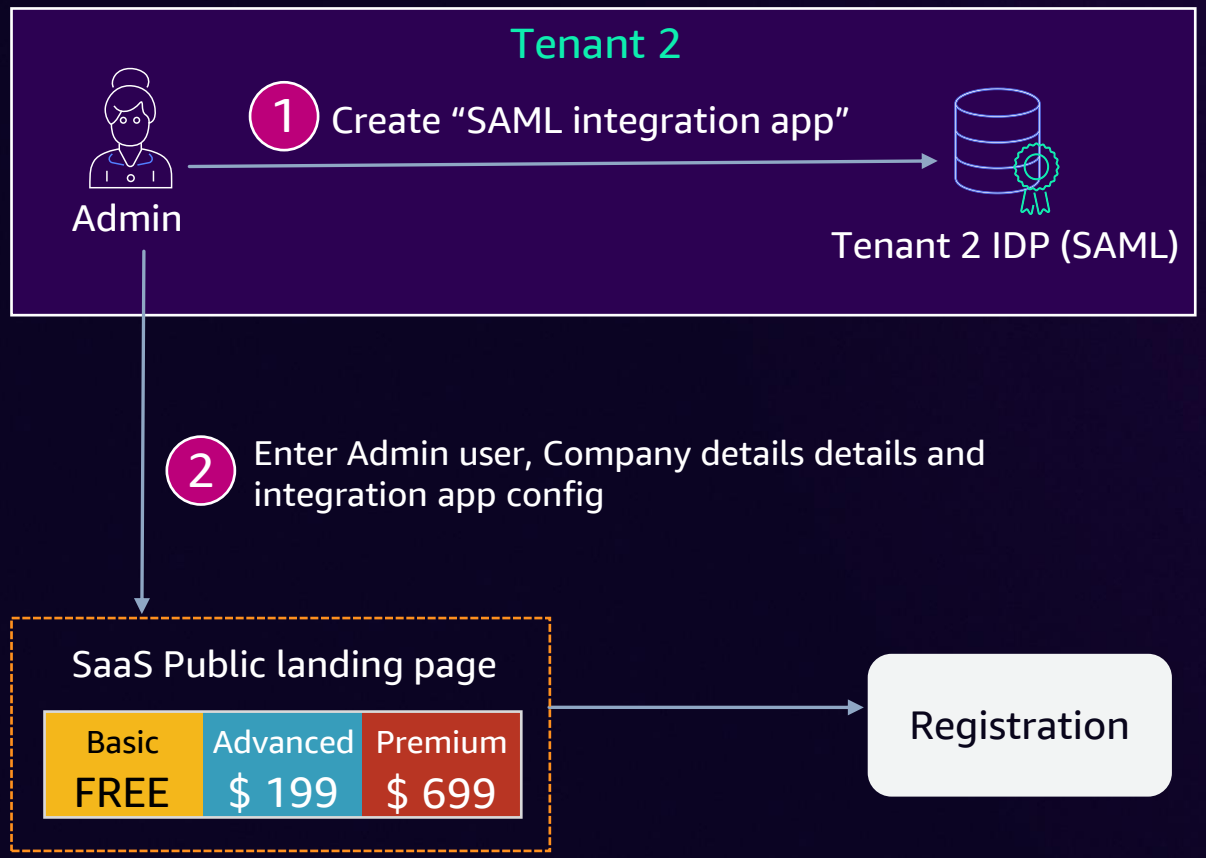
Flow 2 : Federated tenant onboarding



Flow 2 : Federated tenant onboarding

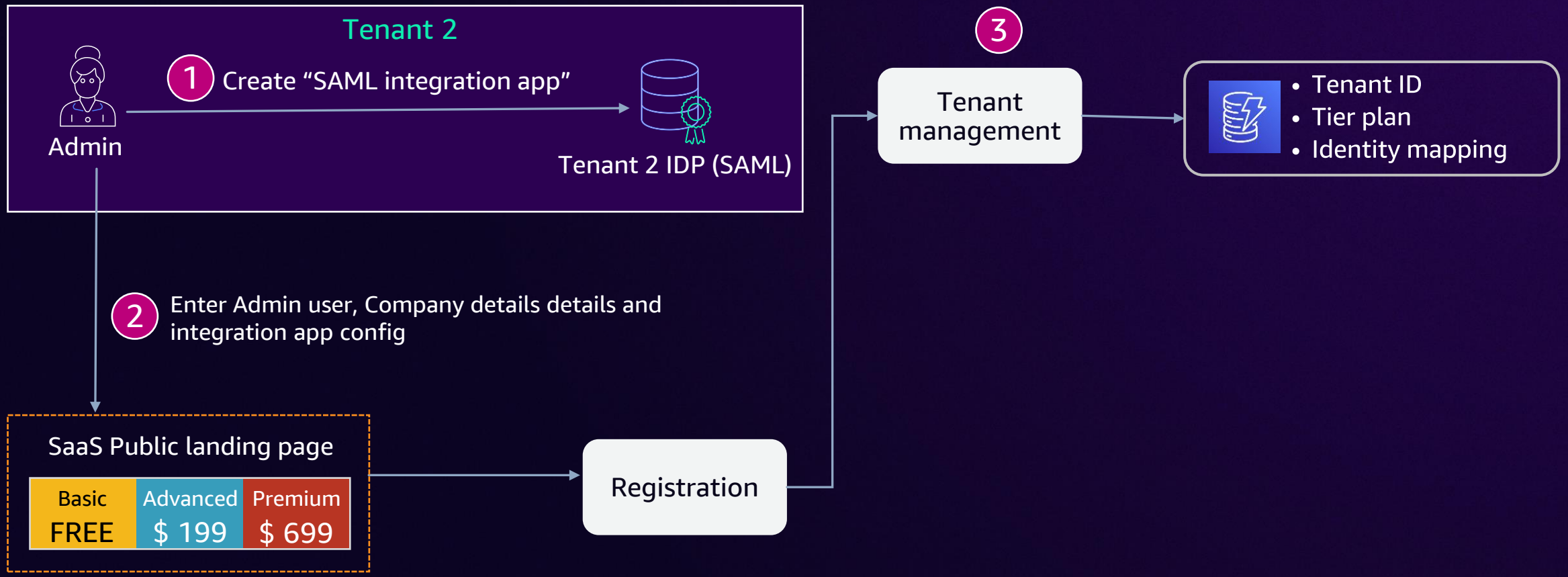


Flow 2 : Federated tenant onboarding

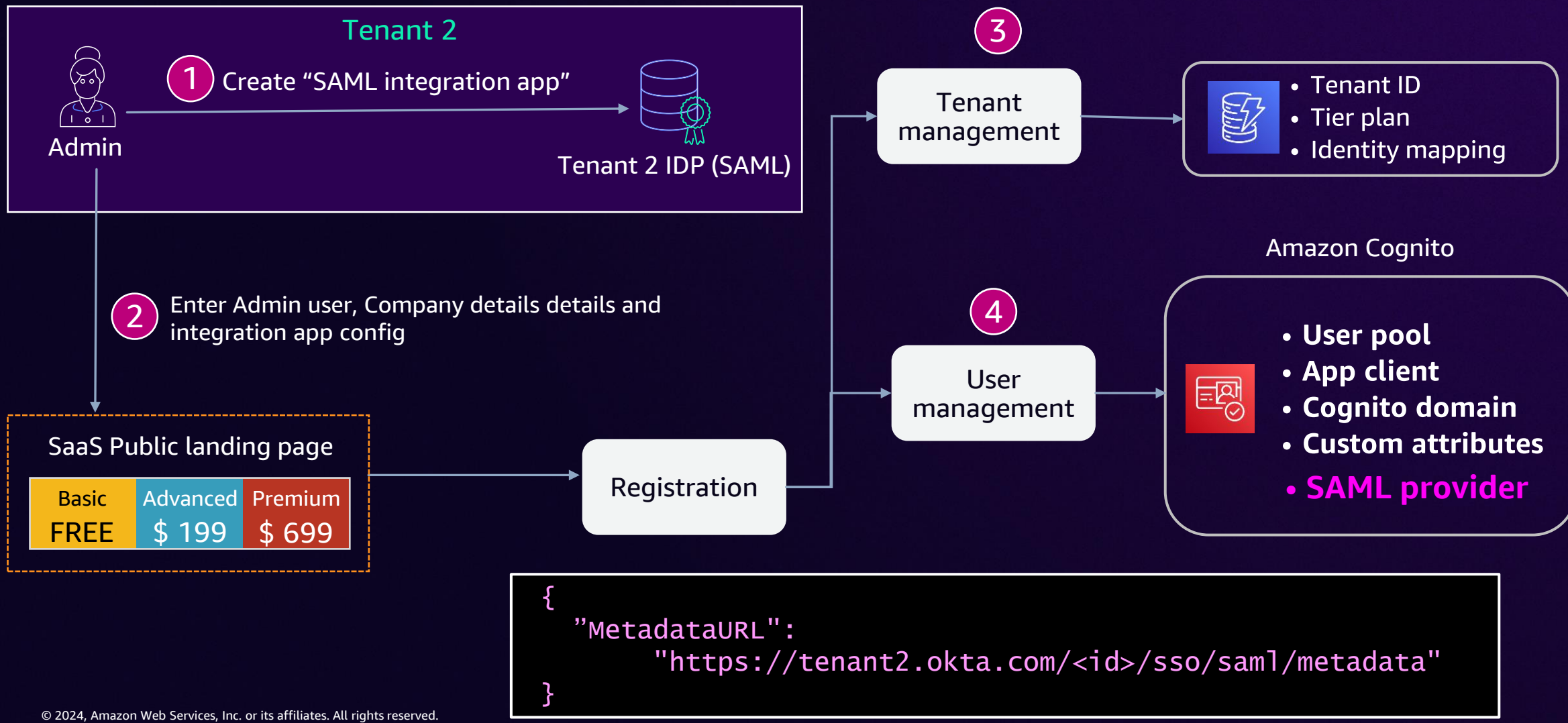



```
{
 ”adminName”: ” Jane Doe ”,
 ”adminEmail”: ”jane@tenant2.com”,
 ”tier”: ”Premium”,
 ”companyName”:”Tenant2”,
 "companyURL":” https://tenant2.com”
 ”MetadataURL”:
        "https://tenant2.okta.com/
              <id>/sso/saml/metadata"
}
```

Flow 2 : Federated tenant onboarding

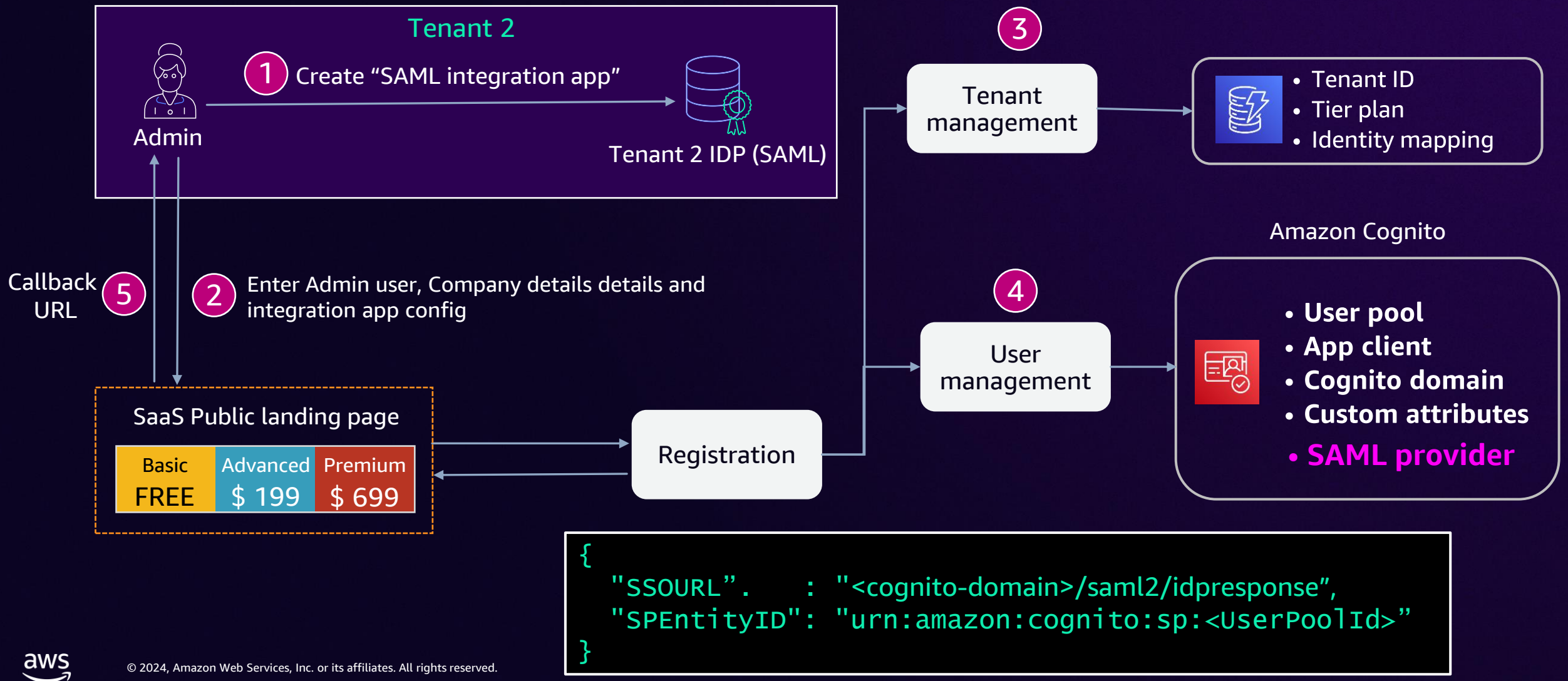


Flow 2 : Federated tenant onboarding



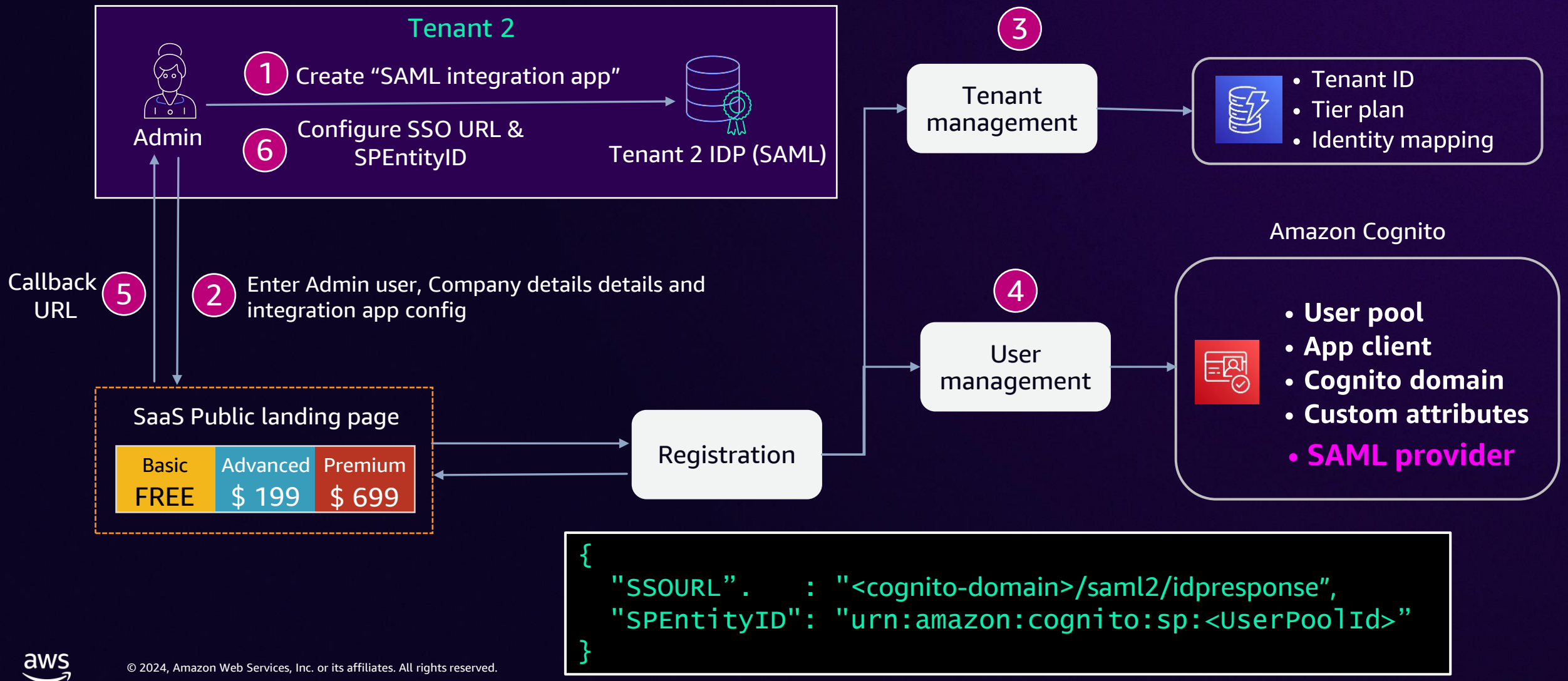

```
{
  ”MetadataURL":
       "https://tenant2.okta.com/<id>/sso/saml/metadata"
}
```

Flow 2 : Federated tenant onboarding




```
{
  "SSOURL".    : "<cognito-domain>/saml2/idpresponse",
  "SPEntityID": "urn:amazon:cognito:sp:<UserPoolId>"
}
```

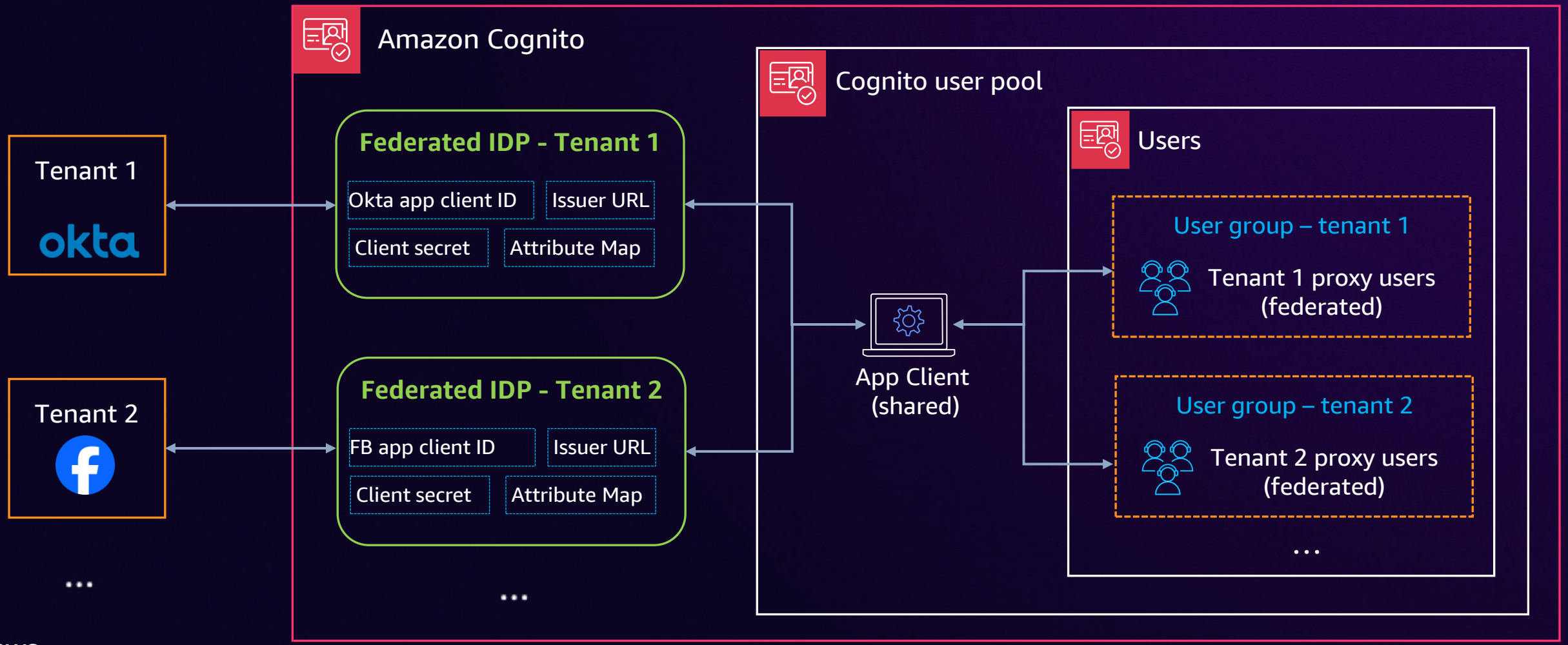
Flow 2 : Federated tenant onboarding




```
{
  "SSOURL".    : "<cognito-domain>/saml2/idpresponse",
  "SPEntityID": "urn:amazon:cognito:sp:<UserPoolId>"
}
```

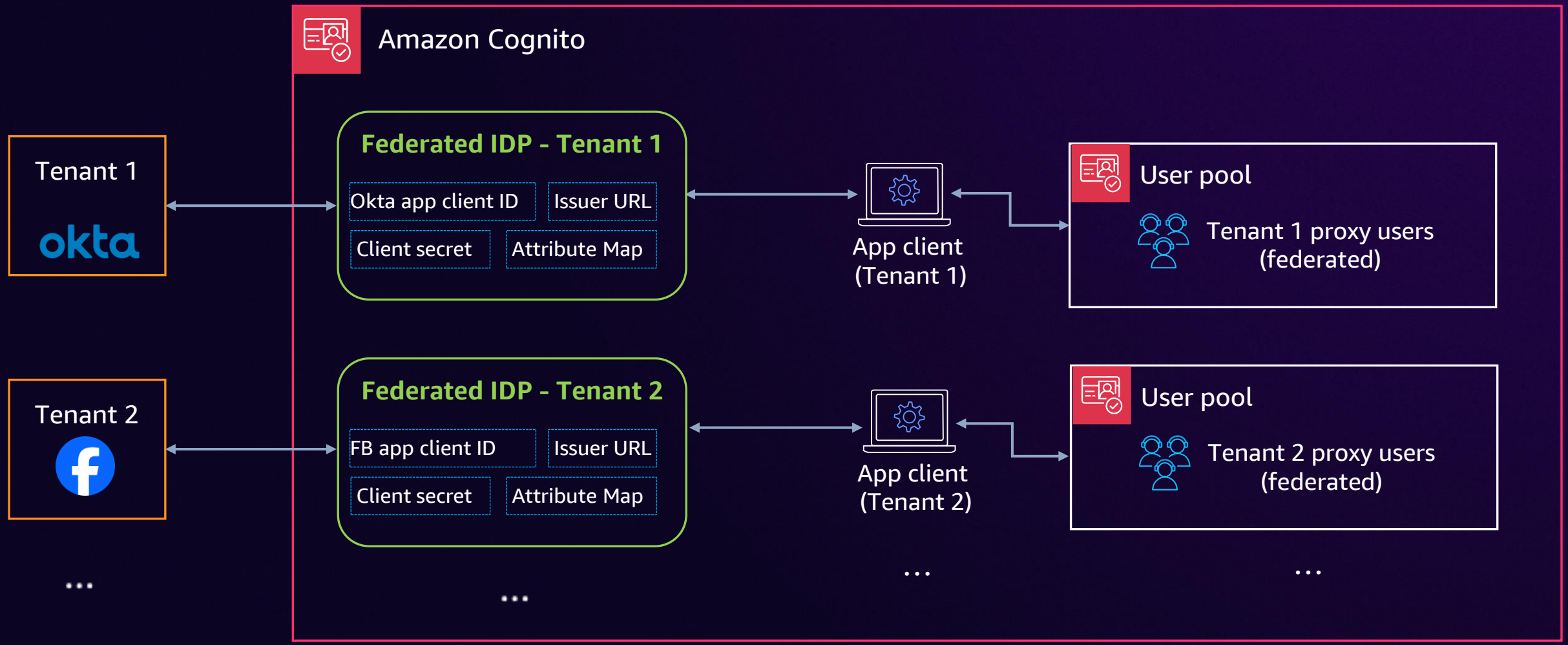
Cognito design – Pool model

Each tenant will have dedicated federated identity provider and shared App client with a shared user pool

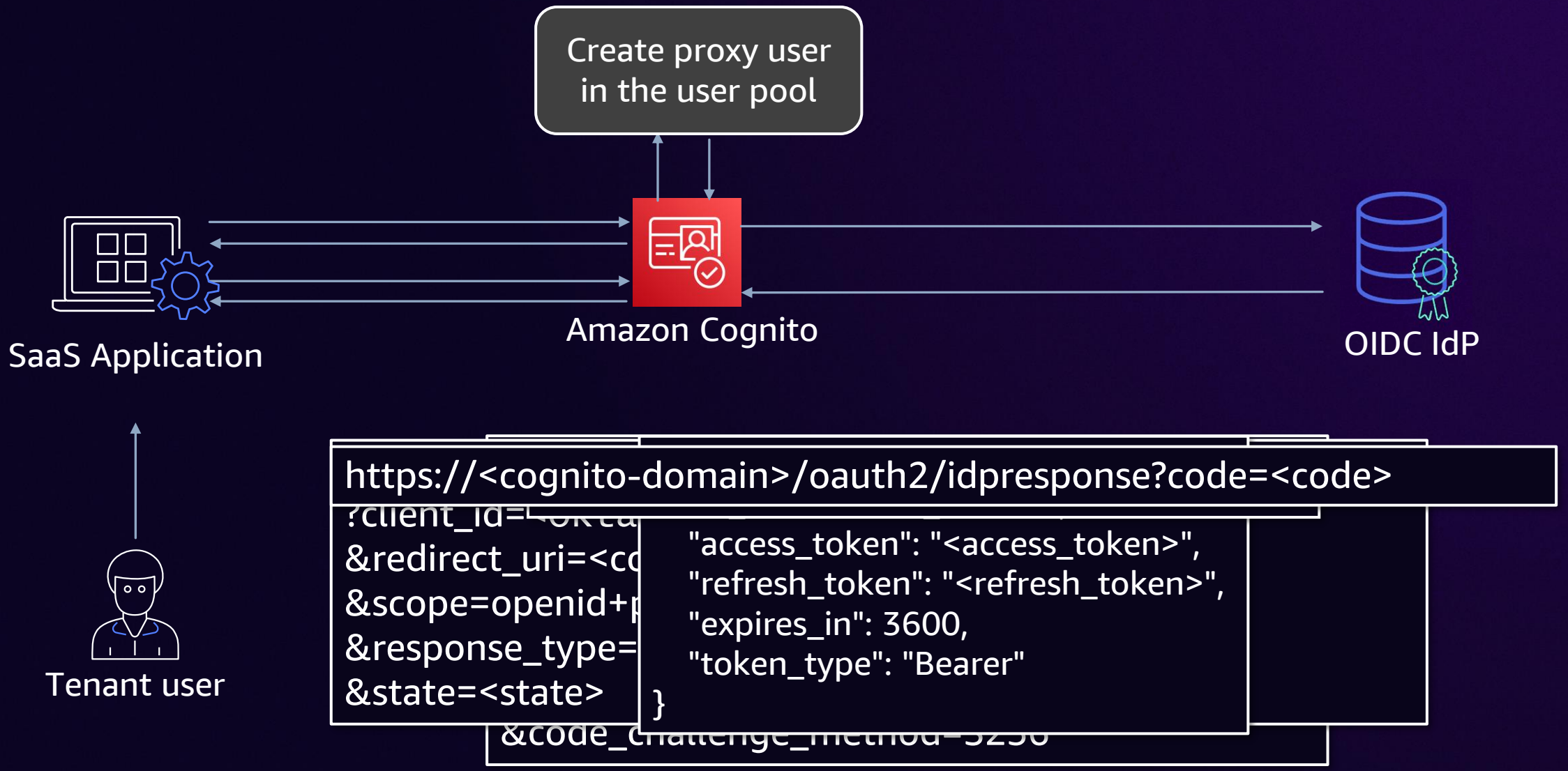


Cognito design – Silo model

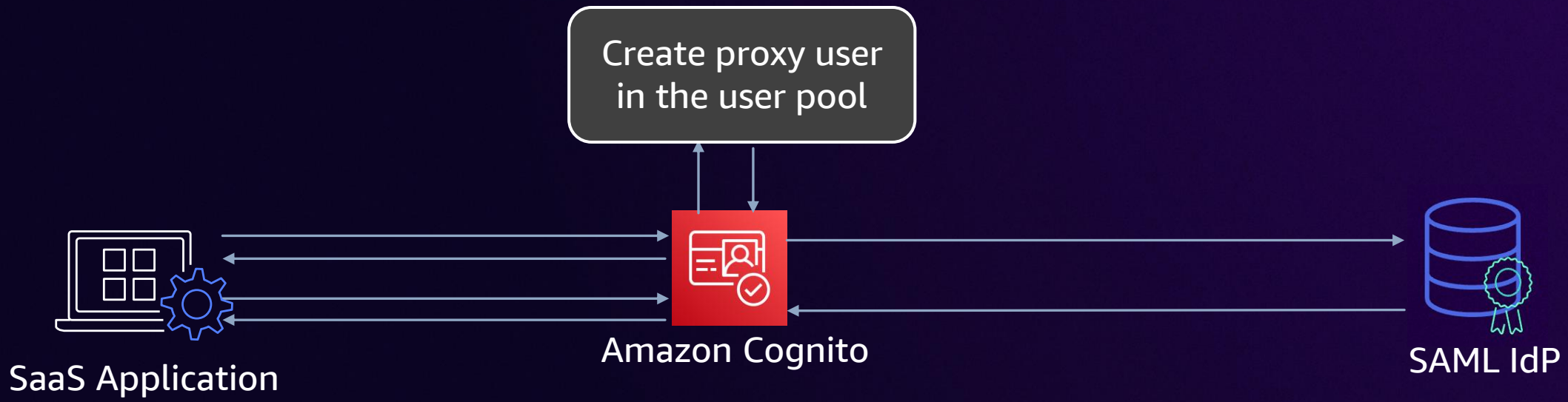
Each tenant will have dedicated federated identity provider and dedicated user pool (and so an app client)



Federated tenant user login – OIDC

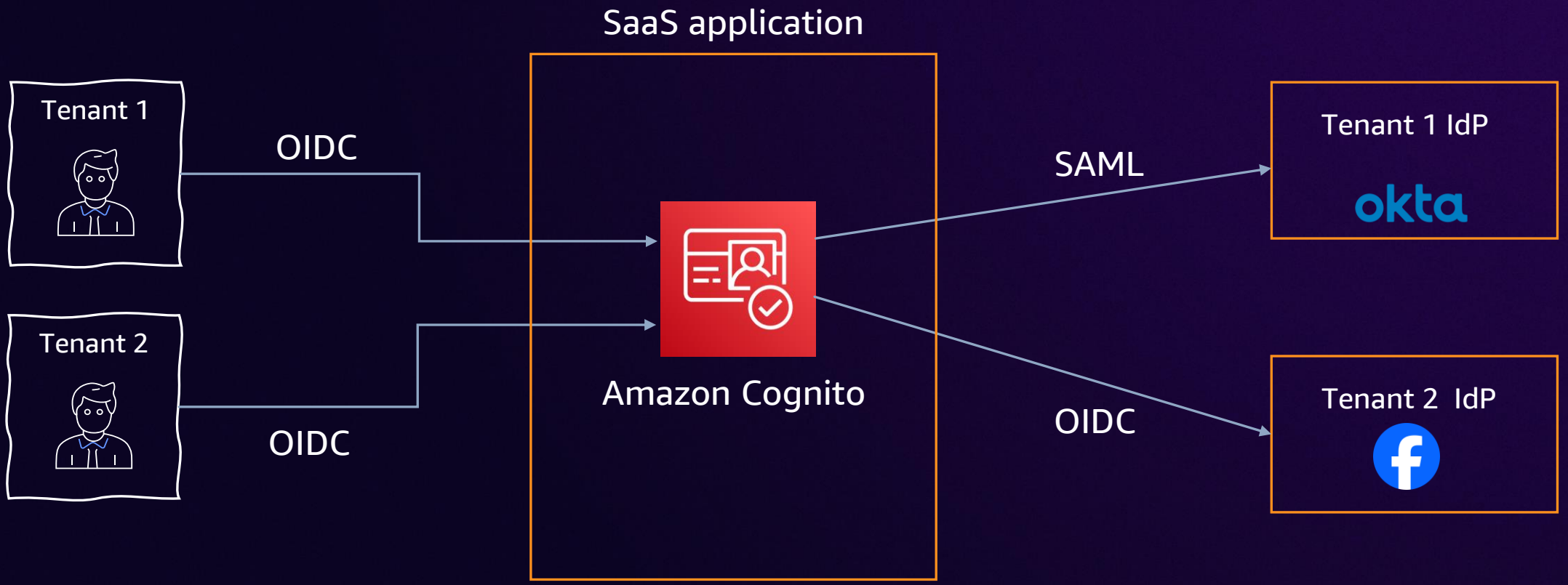


Tenant user login – Federated SAML flow

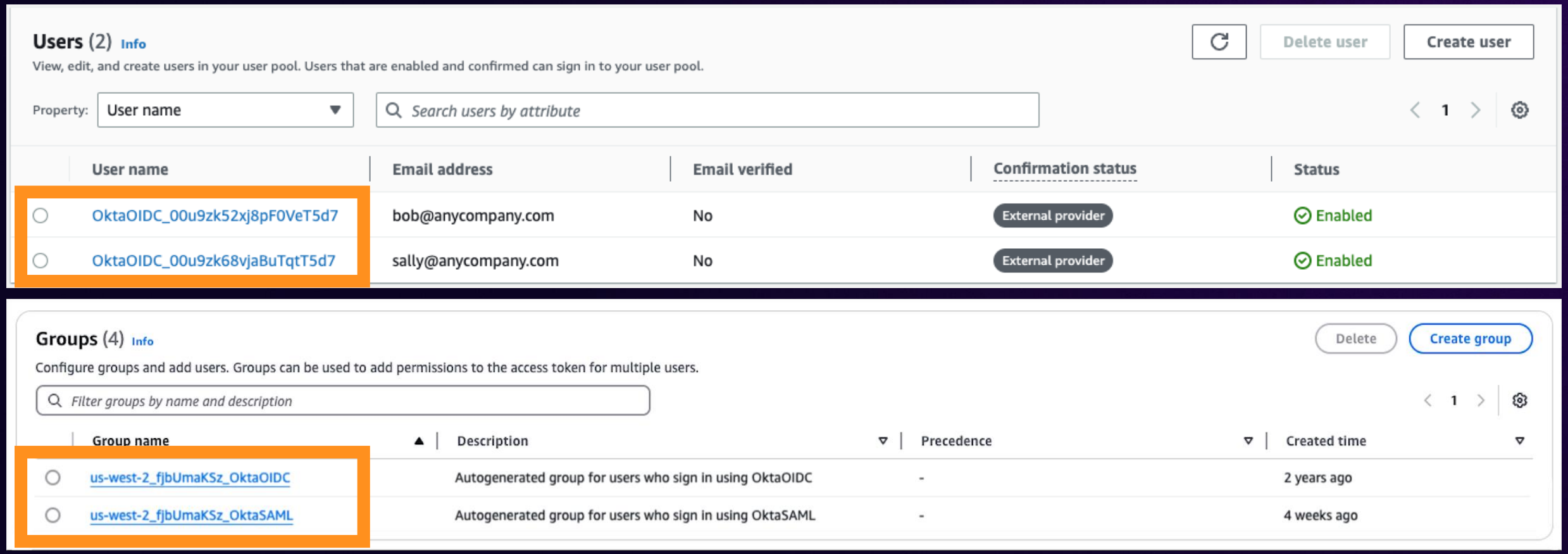



```
<cognito-domain>/oauth2/authorize?client_id=<Cognito Client Id>
&redirect_uri=<SaaS App URL>
&response_type=code
&scope=openid profile email
&code_challenge=<code challenge>
&code_challenge_method=S256
```

Cognito as a federation hub



Federated proxy users



Users (2) Info

View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

Delete user | Create user

Property: User name | Search users by attribute | 1

User name	Email address	Email verified	Confirmation status	Status
OktaOIDC_00u9zk52xj8pF0VeT5d7	bob@anycompany.com	No	External provider	Enabled
OktaOIDC_00u9zk68vjaBuTqtT5d7	sally@anycompany.com	No	External provider	Enabled

Groups (4) Info

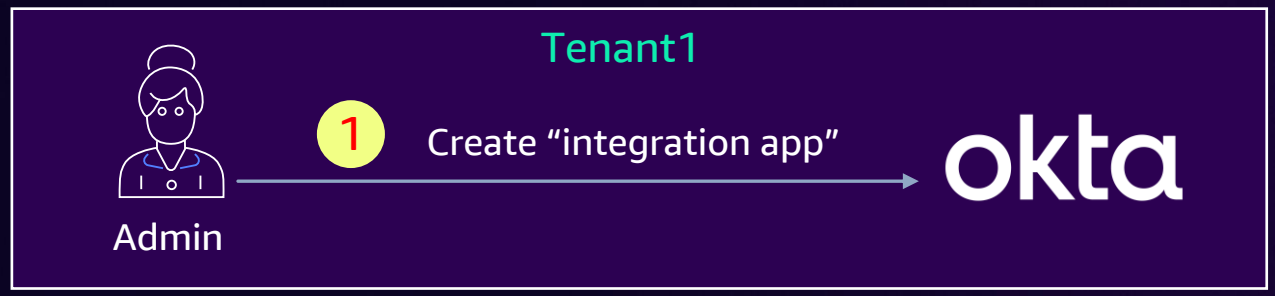
Configure groups and add users. Groups can be used to add permissions to the access token for multiple users.

Delete | Create group

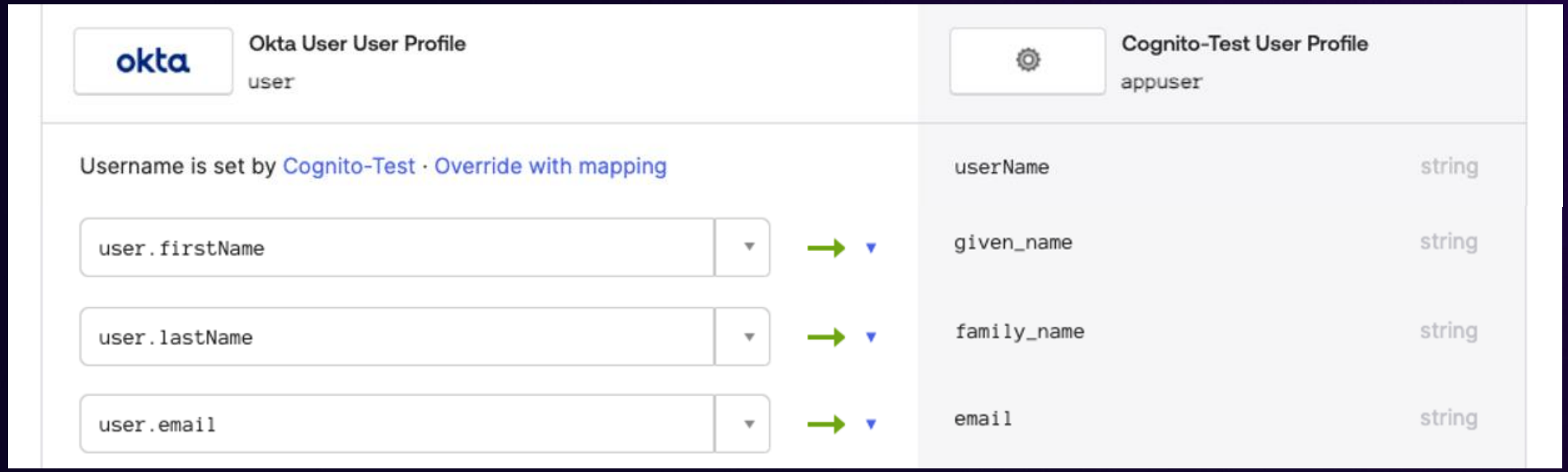
Filter groups by name and description | 1

Group name	Description	Precedence	Created time
us-west-2_fjbUmaKSz_OktaOIDC	Autogenerated group for users who sign in using OktaOIDC	-	2 years ago
us-west-2_fjbUmaKSz_OktaSAML	Autogenerated group for users who sign in using OktaSAML	-	4 weeks ago

Attribute mapping – Tenant IDP

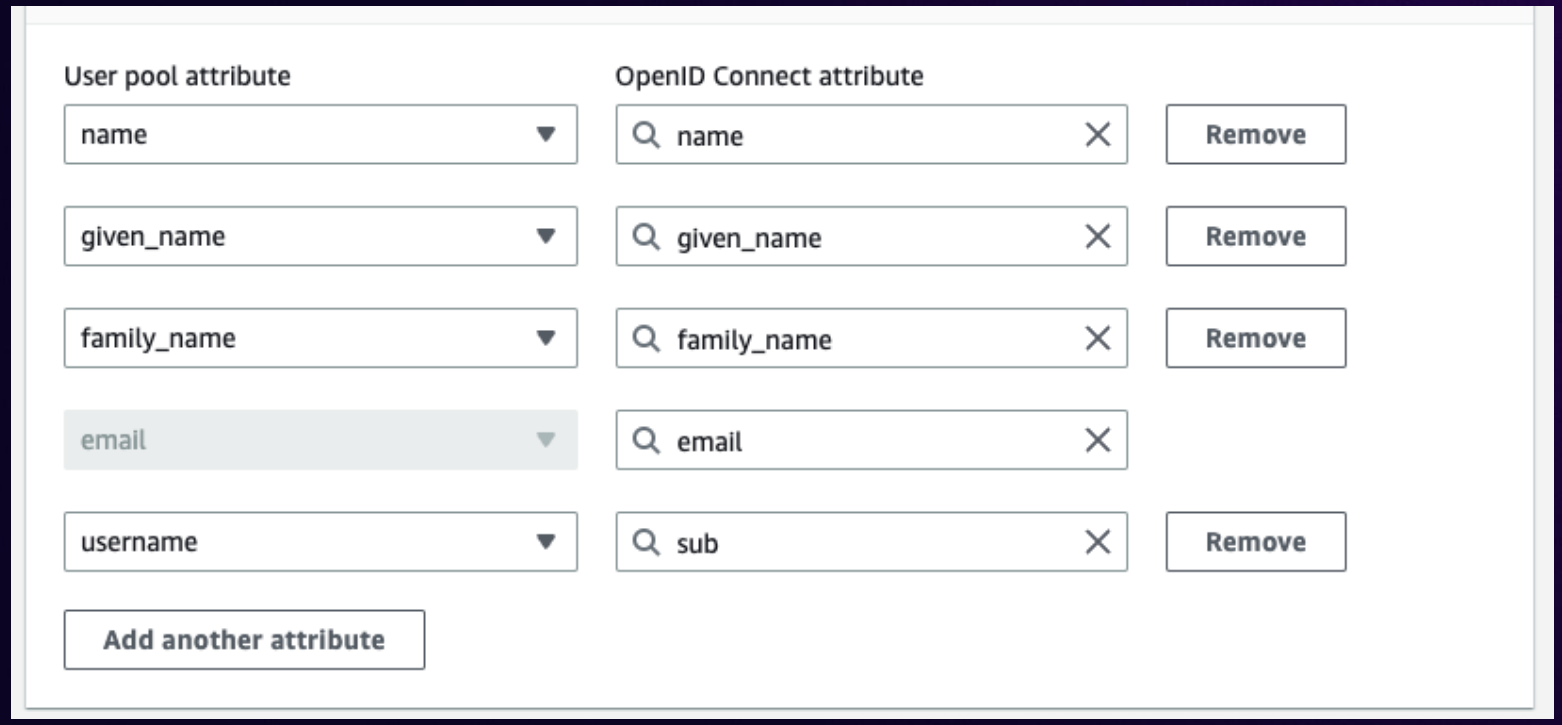


Maps user attributes from tenant IDP to standard OIDC attributes for handshaking



Cognito attribute mappings – OIDC

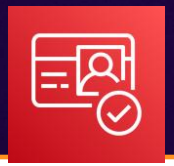
Maps attributes from OIDC ID Token to the Cognito user profile attributes



Cognito attribute mappings – SAML

Name	Name Format	Value
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	Basic	user.email
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	Basic	user.firstName
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	Basic	user.lastName

okta

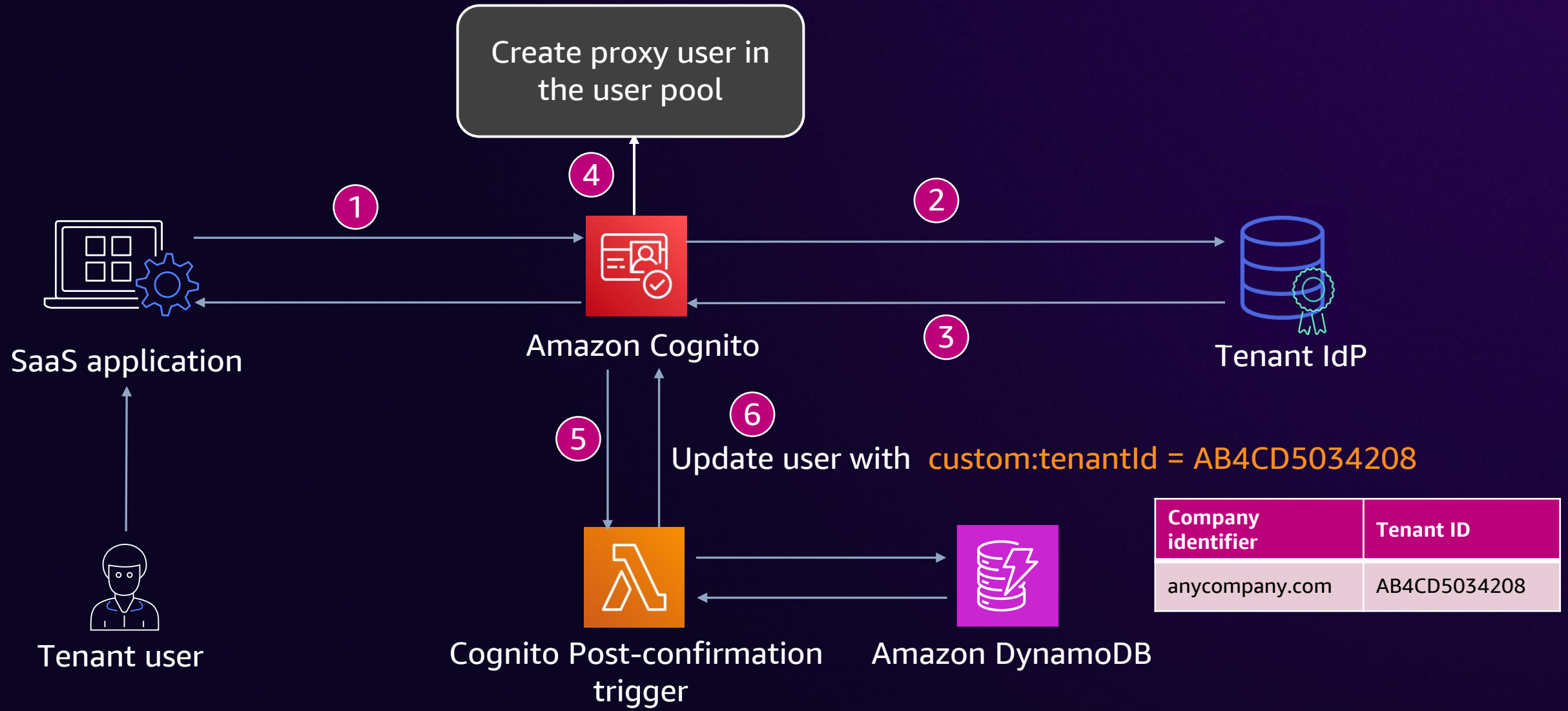


User pool attribute	SAML attribute
email	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
family_name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname
name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name

Cognito Lambda triggers

User Pool Flow	Operation	Description
Custom Authentication Flow	**Define Auth Challenge**	Determines the next challenge in a custom auth flow
	Create Auth Challenge	Creates a challenge in a custom auth flow
	Verify Auth Challenge Response	Determines if a response is correct in a custom auth flow
Authentication Events	Pre authentication Lambda trigger	Custom validation to accept or deny the sign-in request
	Post authentication Lambda trigger	Logs events for custom analytics
	Pre token generation Lambda trigger	Augments or suppresses token claims
Sign-Up	Pre sign-up Lambda trigger	Performs custom validation that accepts or denies the sign-up request
	Post confirmation Lambda trigger	Adds custom welcome messages or event logging for custom analytics
	Migrate user Lambda trigger	Migrates a user from an existing user directory to user pools
Messages	Custom message Lambda trigger	Performs advanced customization and localization of messages
Token Creation	Pre token generation Lambda trigger	Adds or removes attributes in Id tokens
Email and SMS third-party providers	Custom sender Lambda triggers	Uses a third-party provider to send SMS and email messages

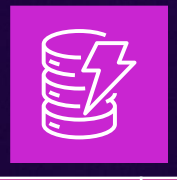
Adding the tenant context and custom attributes



Company identifier	Tenant ID
anycompany.com	AB4CD5034208

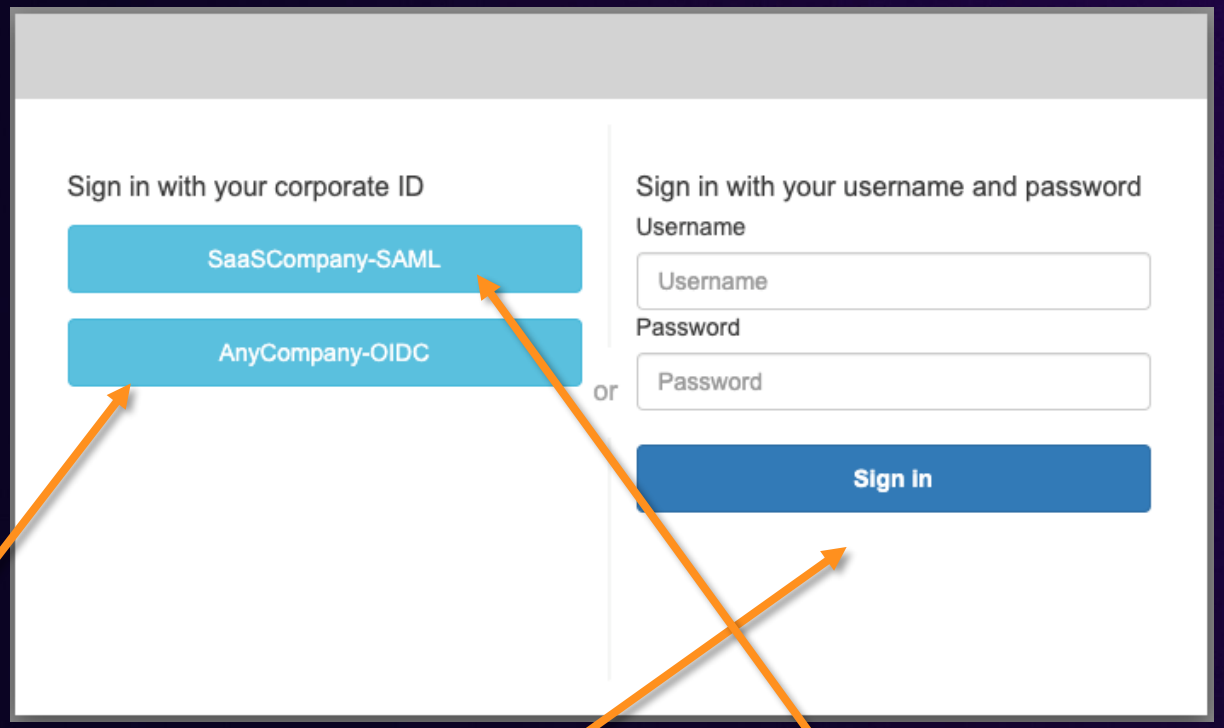
Inside the Lambda trigger

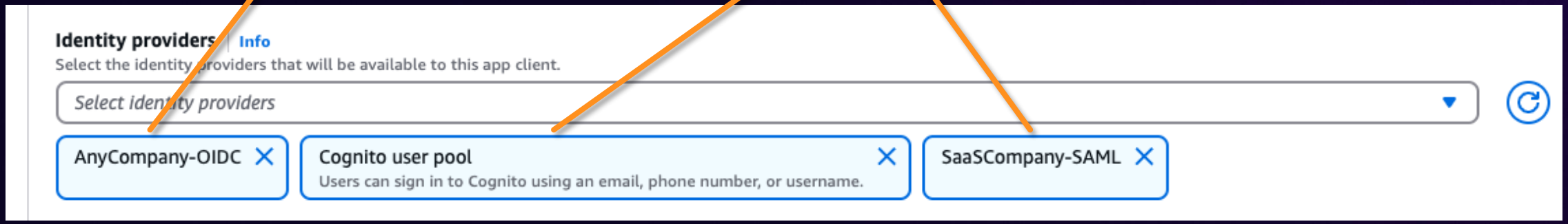
```
export const handler = async (
  event: PostConfirmationTriggerEvent,
  context: Context
): Promise<any> => {
  const email = event.request.userAttributes.email;
  const tenantId = await DynamoDB.fetchTenantId(email);
  const client = new CognitoIdentityProviderClient({ region: process.env.AWS_REGION });
  const input = {
    UserPoolId: event.userPoolId,
    Username: event.userName,
    UserAttributes: [
      {
        Name: 'custom:tenant-id',
        Value: tenantId,
      },
    ],
  };
  const command = new AdminUpdateUserAttributesCommand(input);
  await client.send(command);
  return event;
};
```



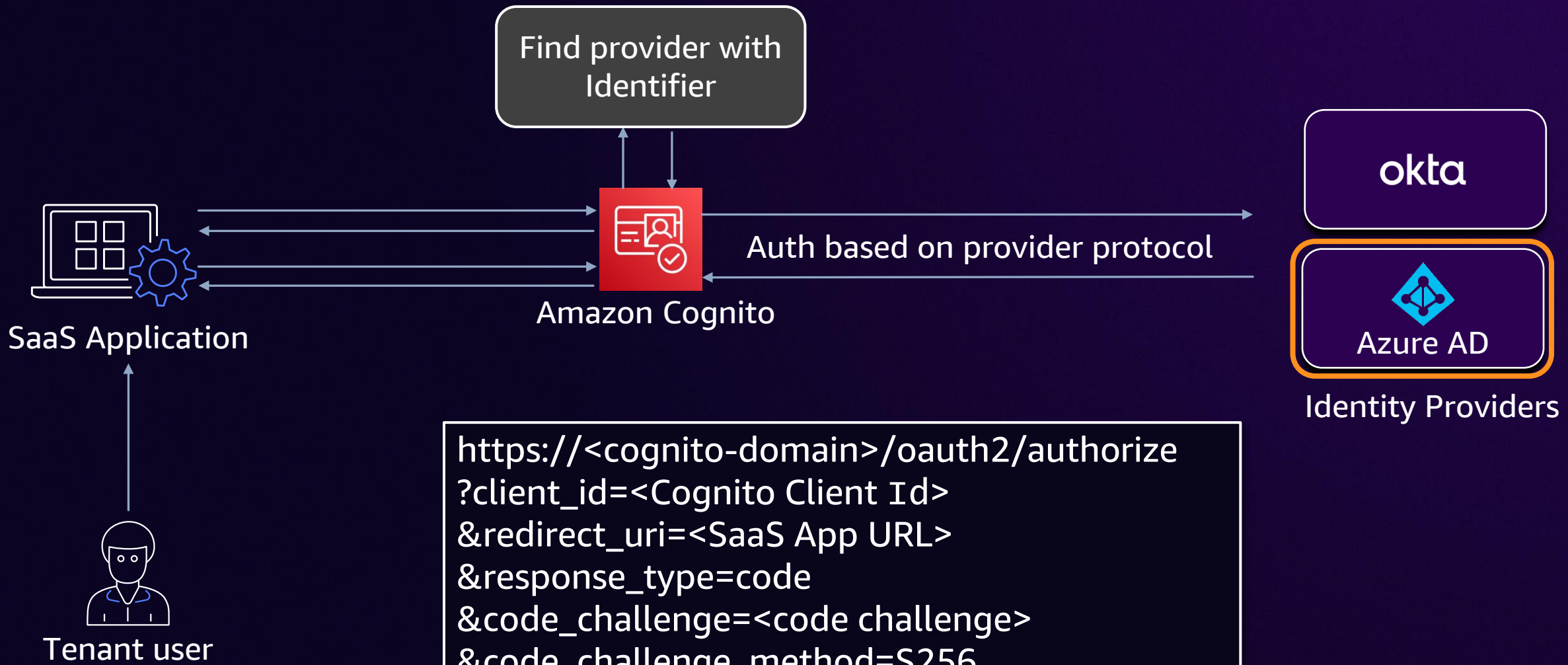
Company identifier	Tenant ID
saascompany.com	AB4CD5034208
anycompany.com	06EC0C83E010

User sign-in – Default behavior

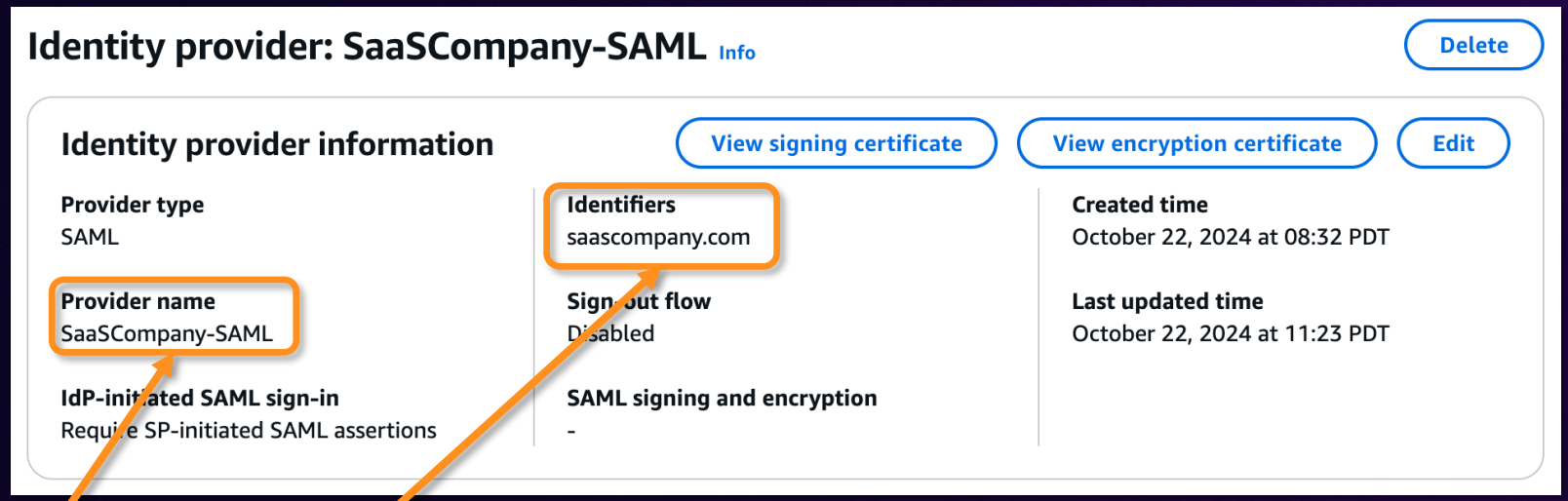




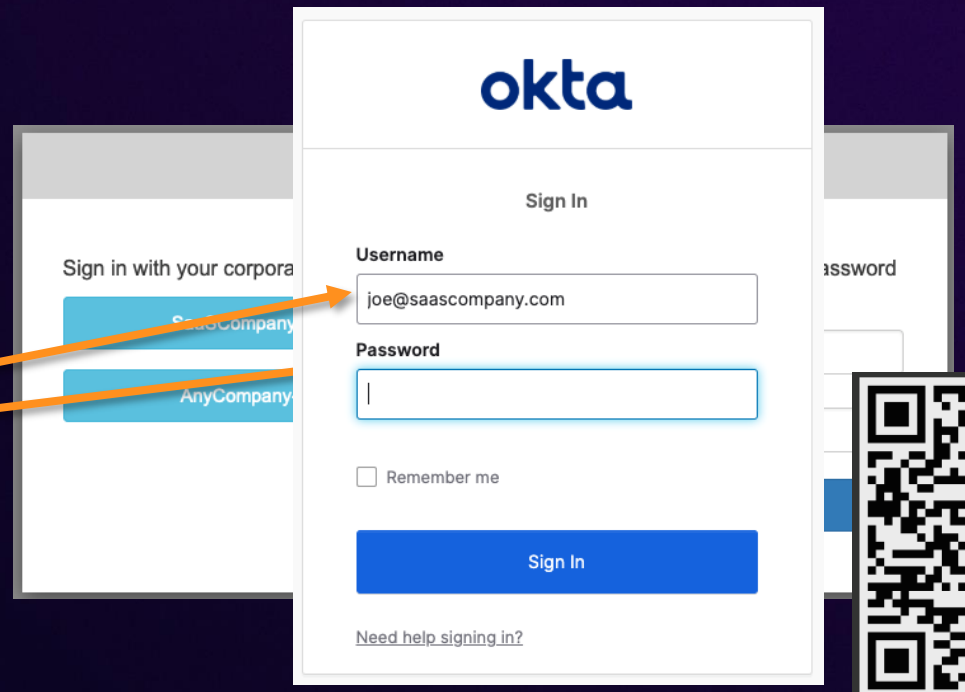
Dynamically select the correct provider



Configure your client

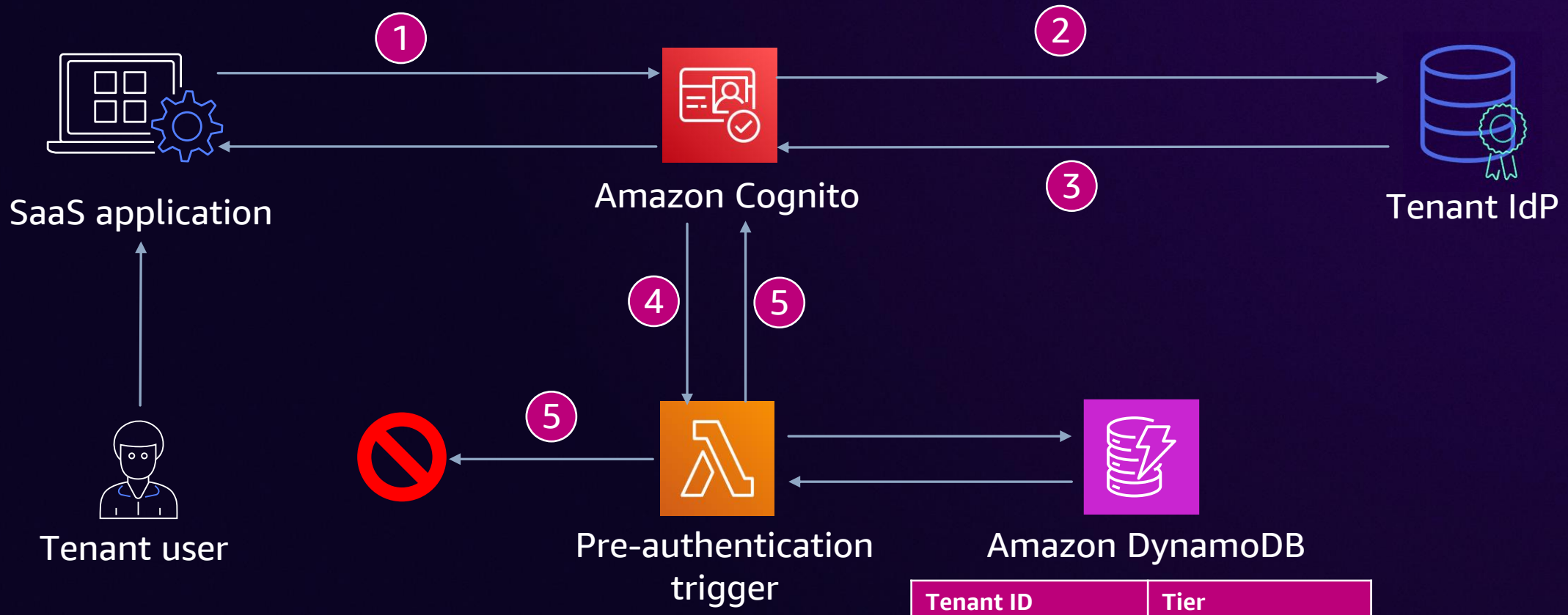


https://<cognito-domain>/oauth2/authorize
?client_id=<Cognito Client Id>
&redirect_uri=<SaaS App URL>
&response_type=code
&code_challenge=<code challenge>
&code_challenge_method=S256
&idp_identifier=saascompany.com
&identity_provider=SaaSCompany-SAML
&login_hint=joe@saascompany.com



Special considerations

Handling entitlements



Tenant ID	Tier
AB4CD5034208	Platinum

Federated considerations with Cognito

- A deeper understanding of OAuth 2.0/OIDC is helpful for understanding Cognito's mental model
- Quotas (both resource and rate) are in effect
- Price goes up a bit
  - SAML/OIDC federation MAU cost is more than user pool/social
- Cognito identity pools are NOT at play here

Quotas

Pricing

Summary

- Federation has more moving parts, but is automatable
- Cognito is a federation hub
- Lambda triggers for customization
- Reconcile user differences with mapping
- Automatically pick federated IdP when pooled
- Unify federated and unfederated in a single onboarding process

Thank you!

Please complete the session survey in the mobile app

Toby Buckley
tobuck@amazon.com

Dhammika Sriyananda
sriyana@amazon.com