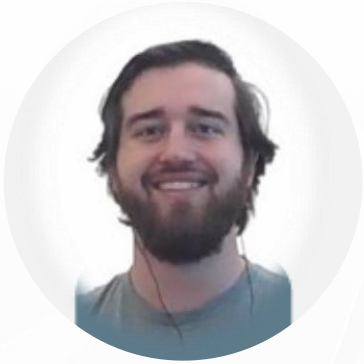
OAuth 2.0 in Healthcare

Presenters



Michael Briggs

Interconnect R&D

Takeaways First

“OAuth 2.0 is worth the investment.”

– *Someone who has implemented OAuth 2.0*

Agenda

01 Why Invest

02 Before OAuth 2.0

03 What is OAuth 2.0?

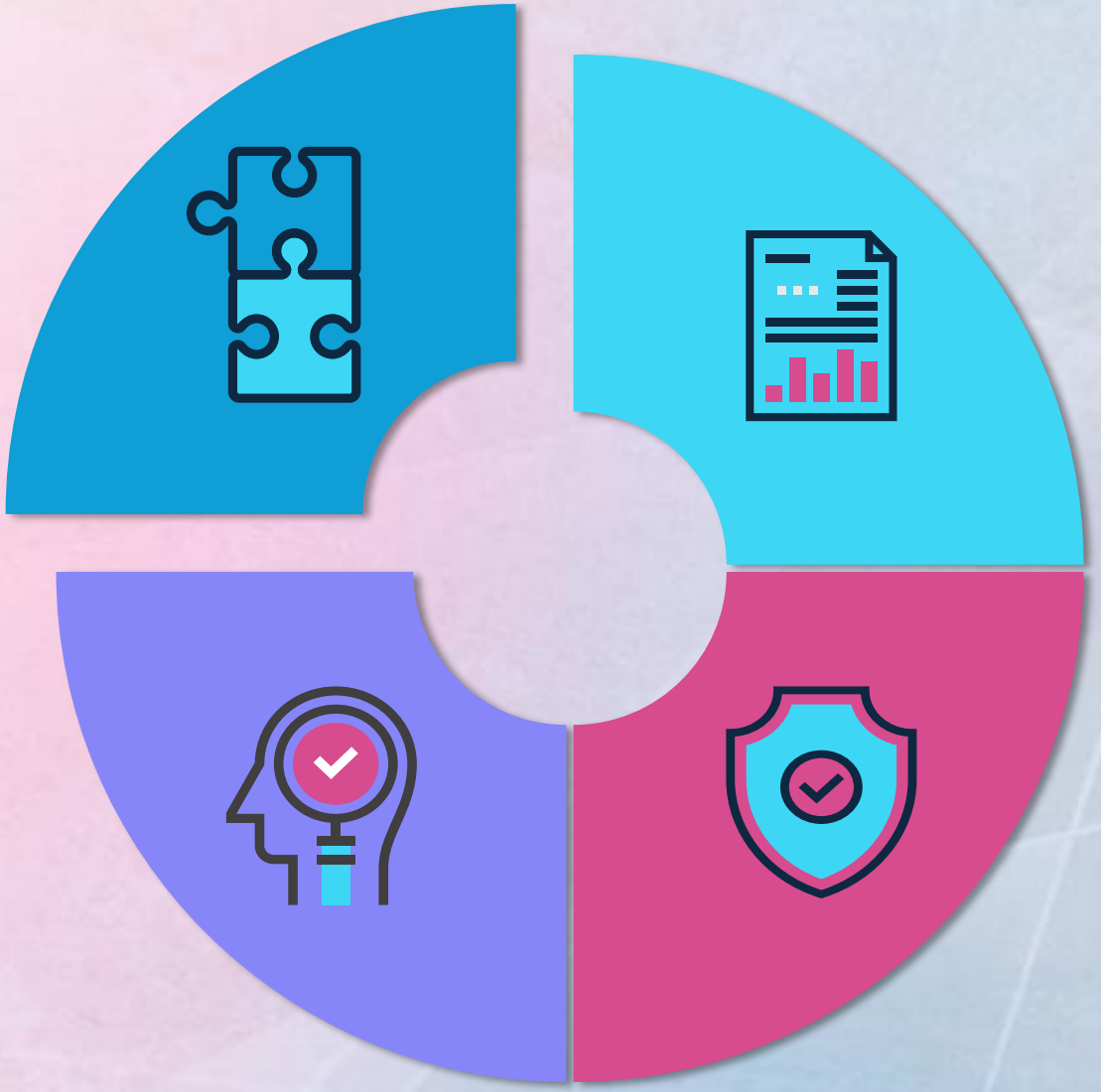
04 Unpacking Each Piece

05 Security Deep Dive

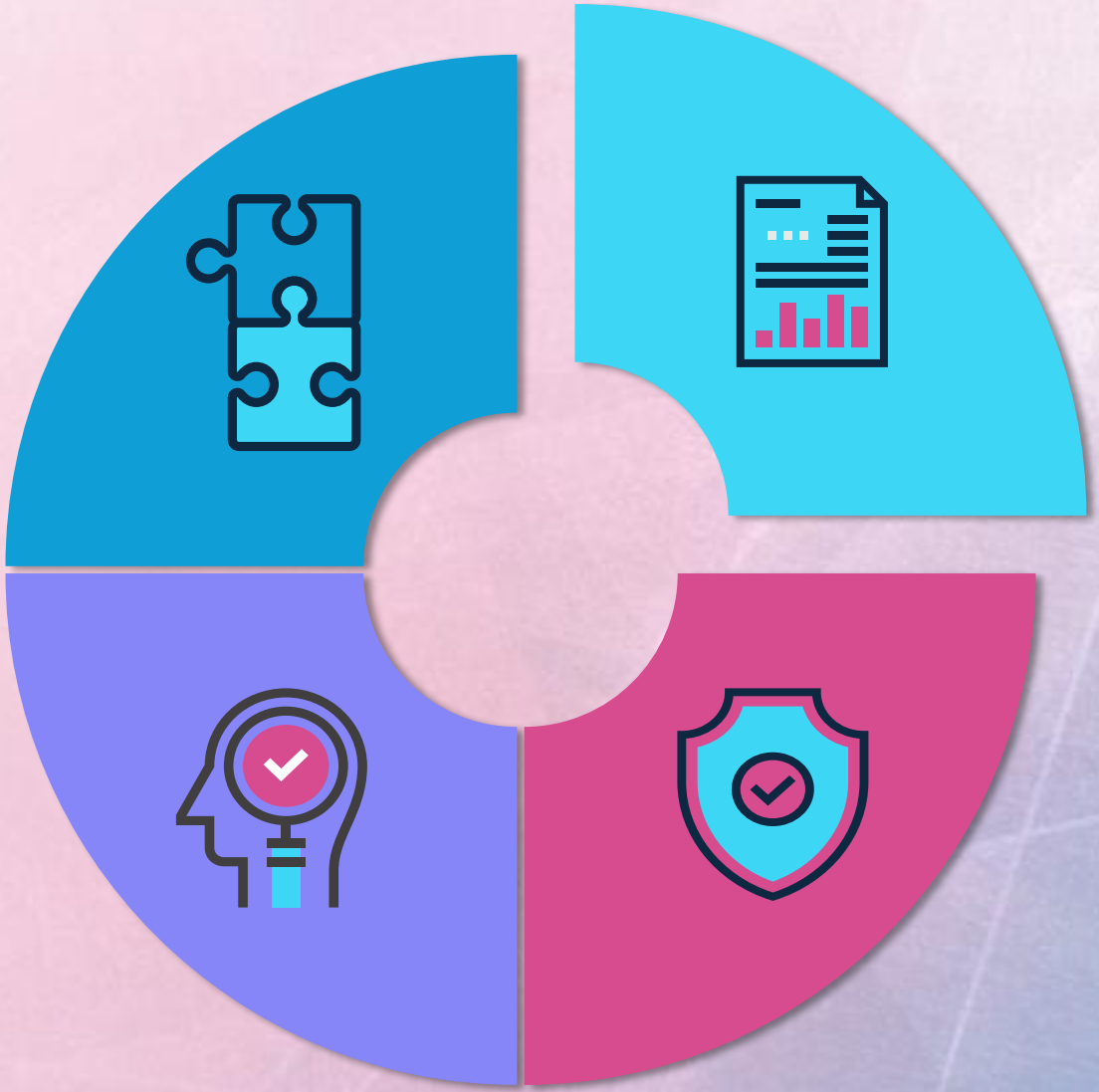
06 Wrap Up

Why Invest

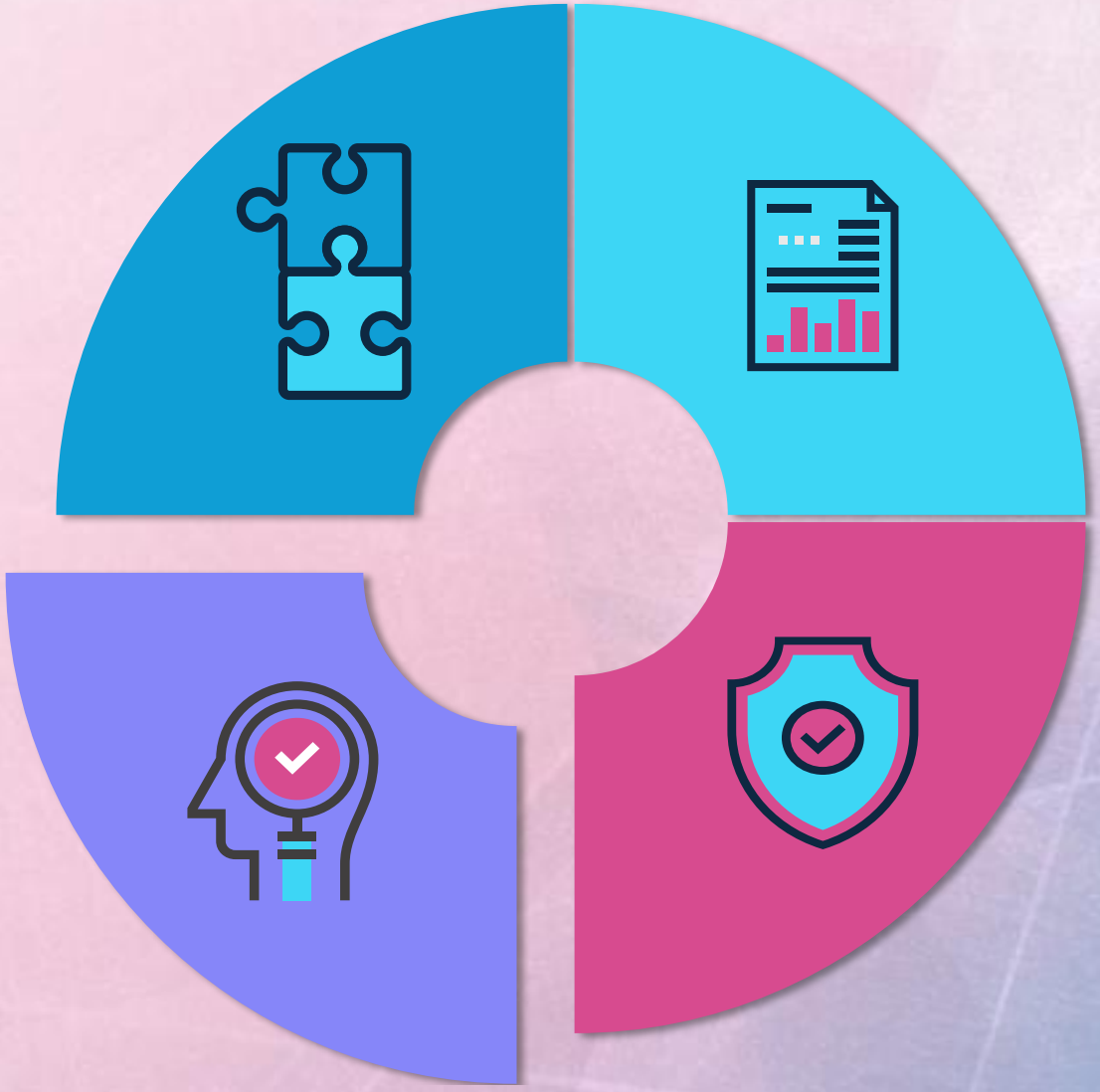
Seamless Experience



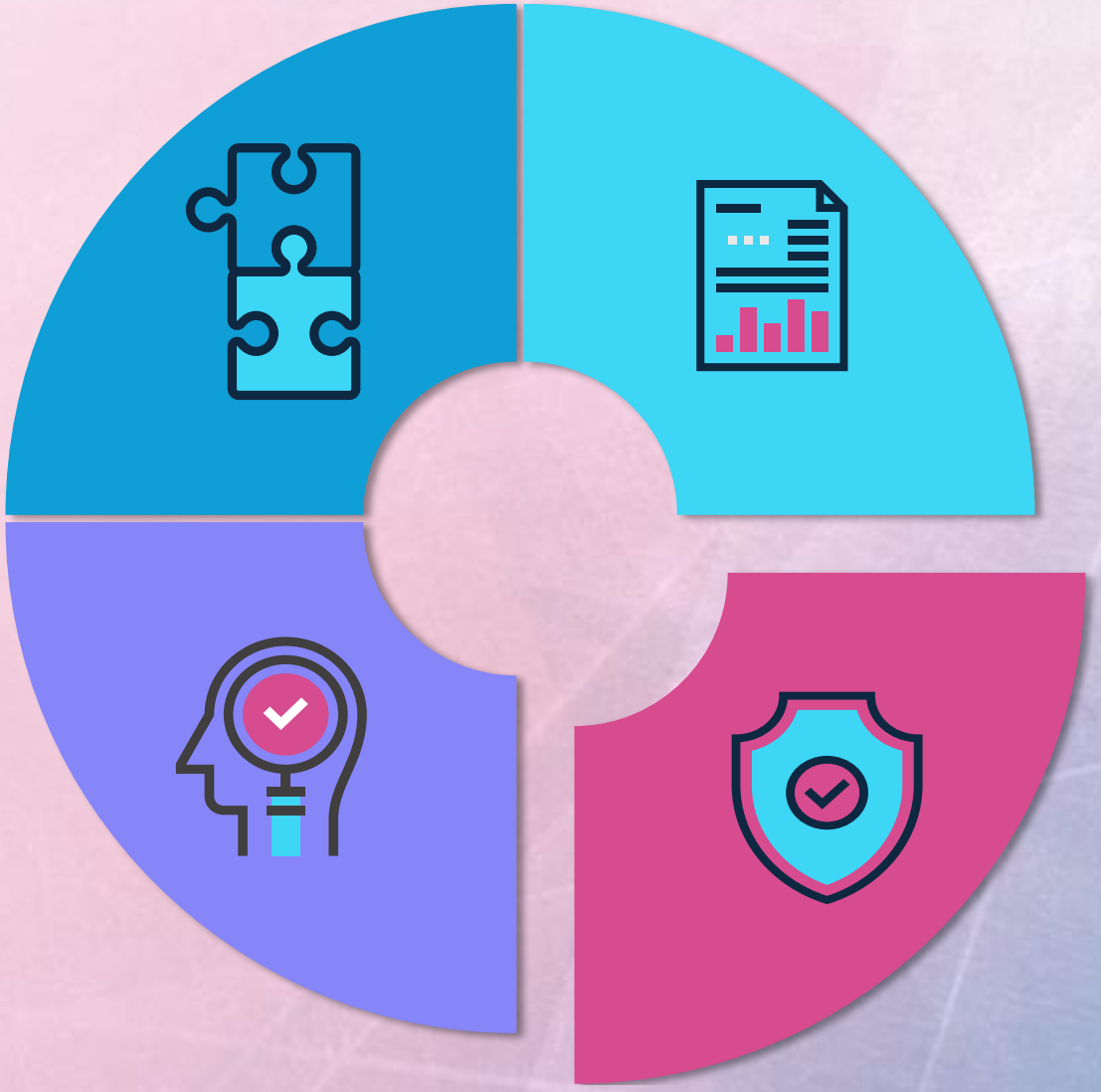
Data Transparency



Easy to use



Protection



Before OAuth 2.0

The People



Alice



Bob



Sue

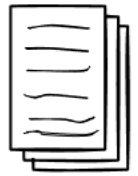


Billy

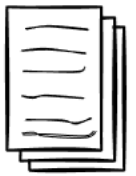
The Data



Alice's Records



Bob's Records

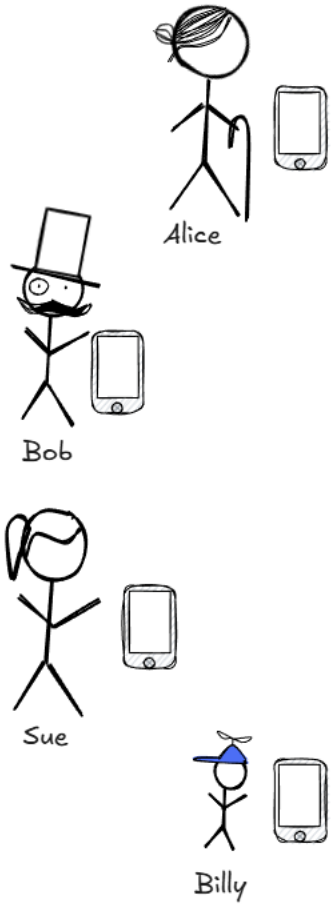


Sue's Records

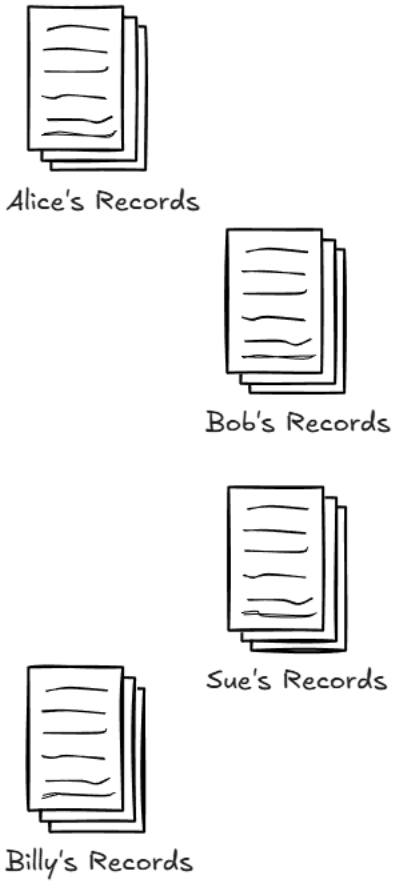


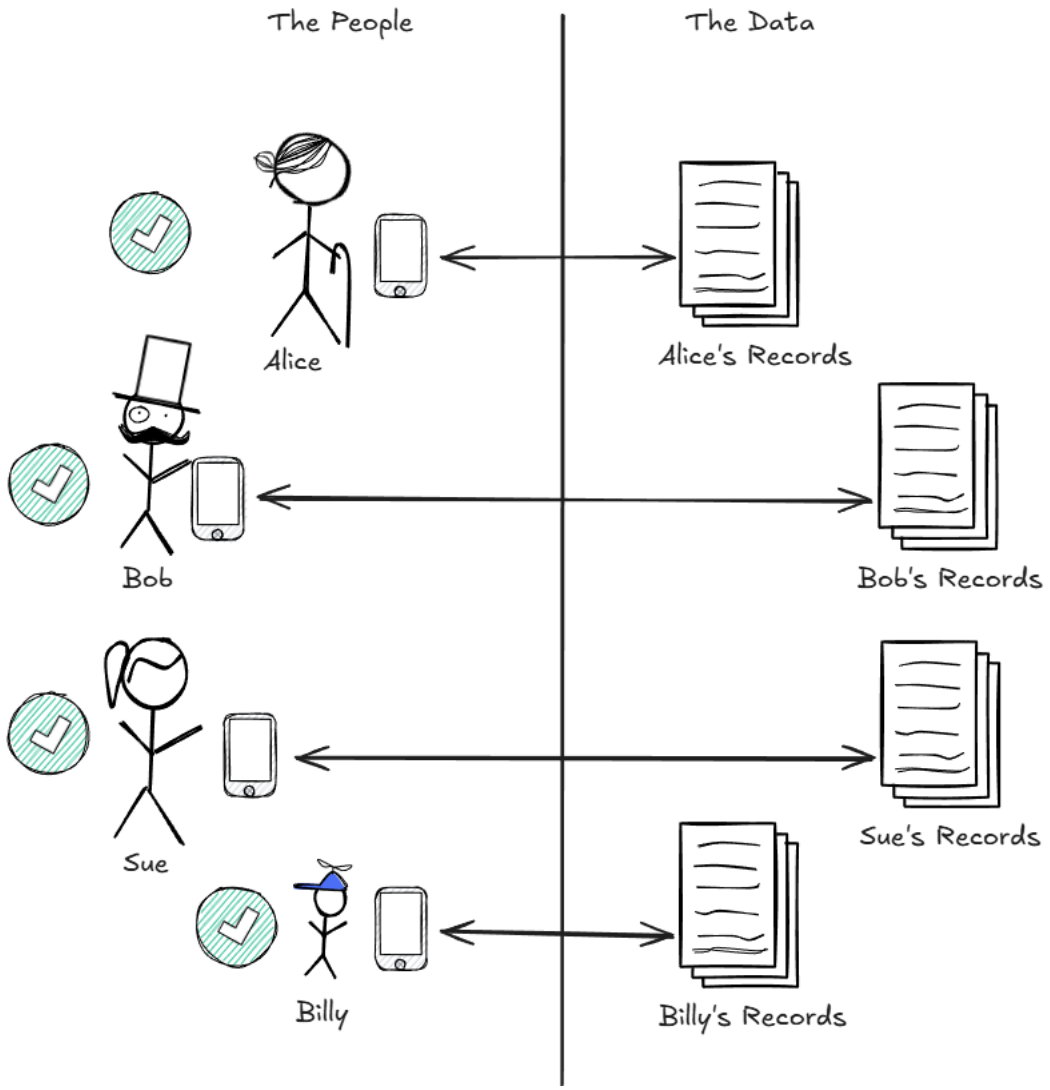
Billy's Records

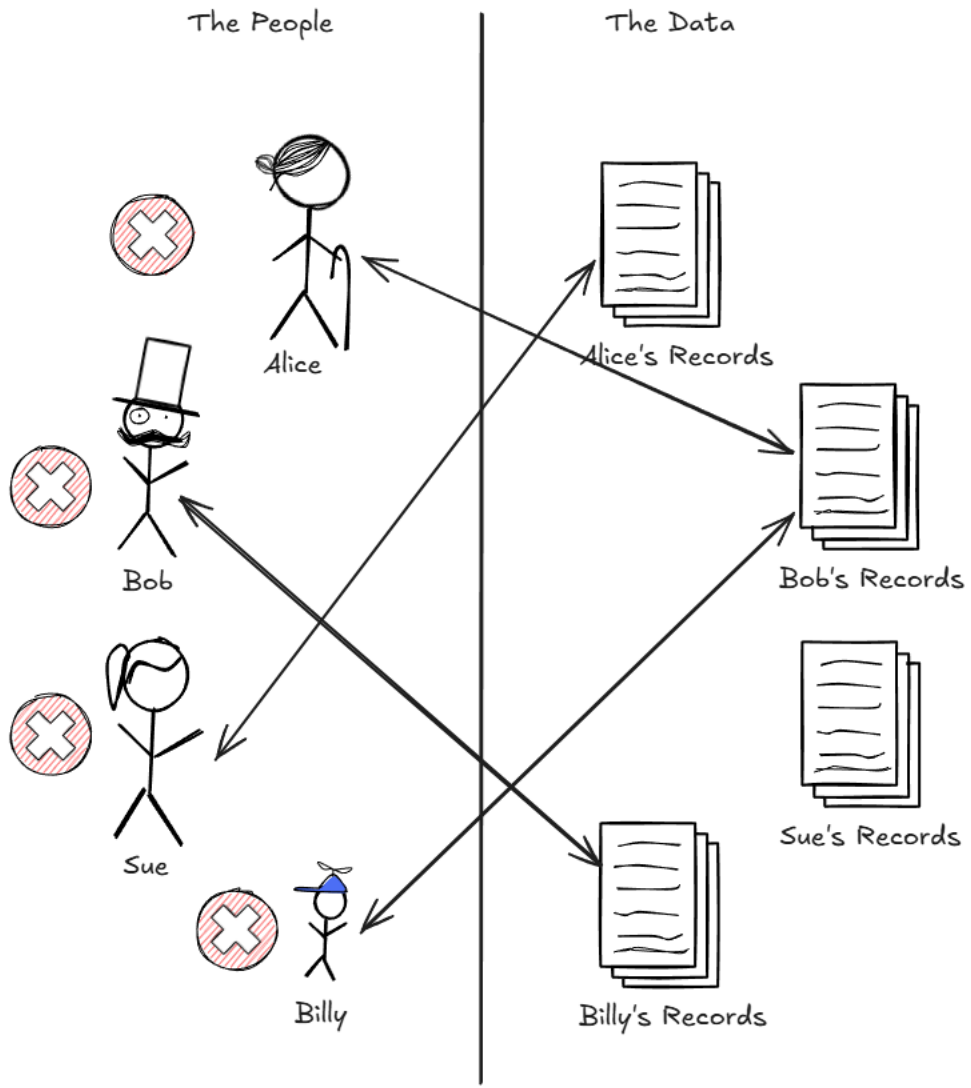
The People

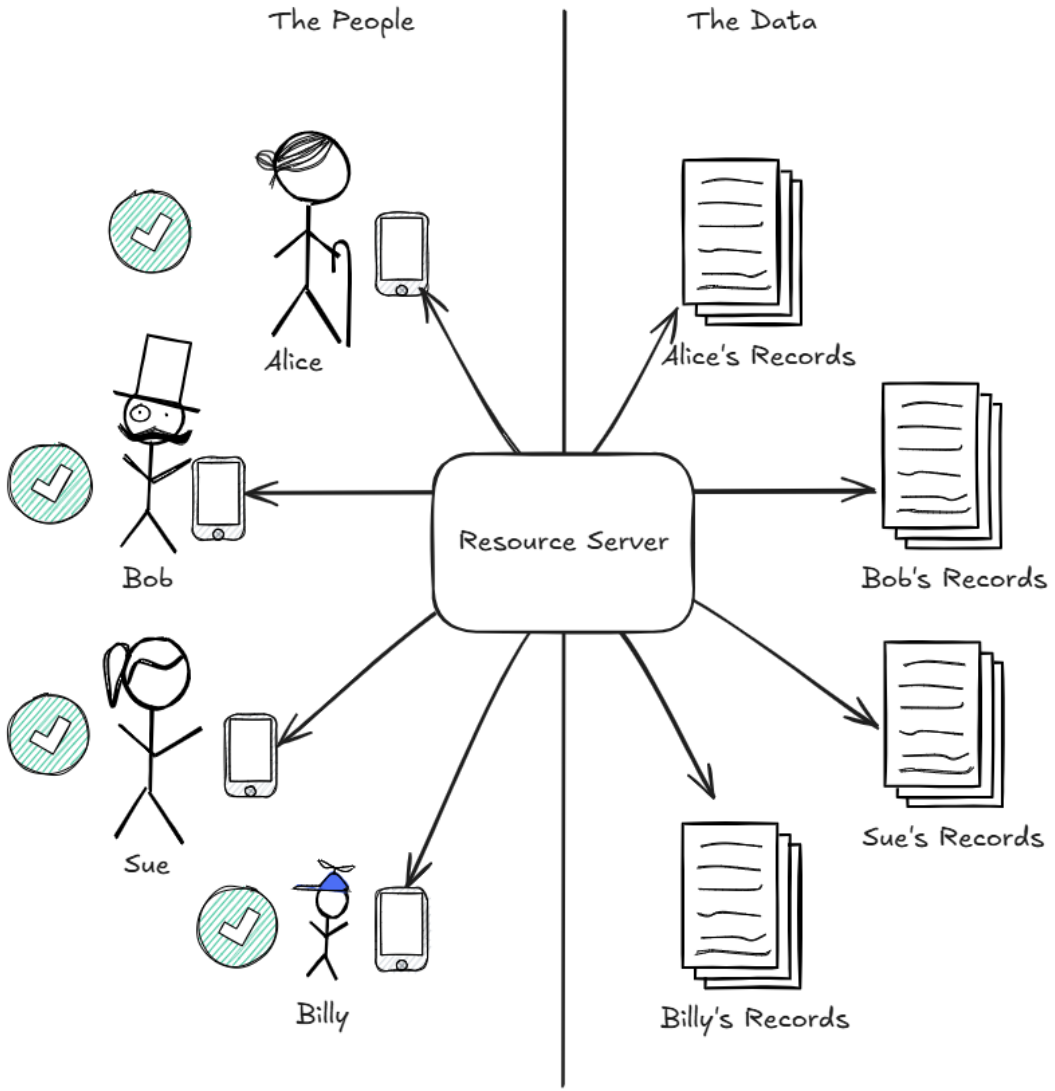


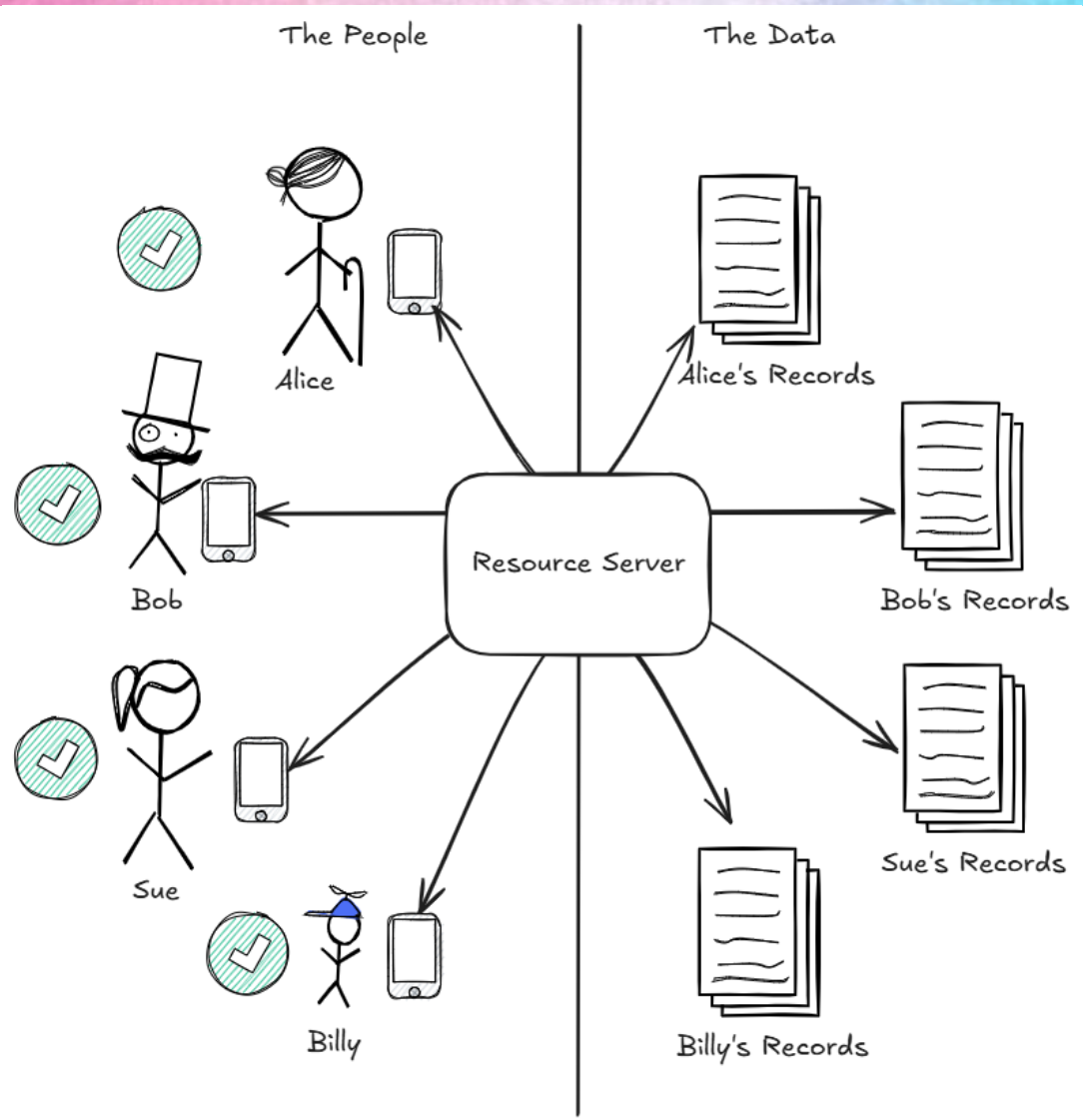
The Data



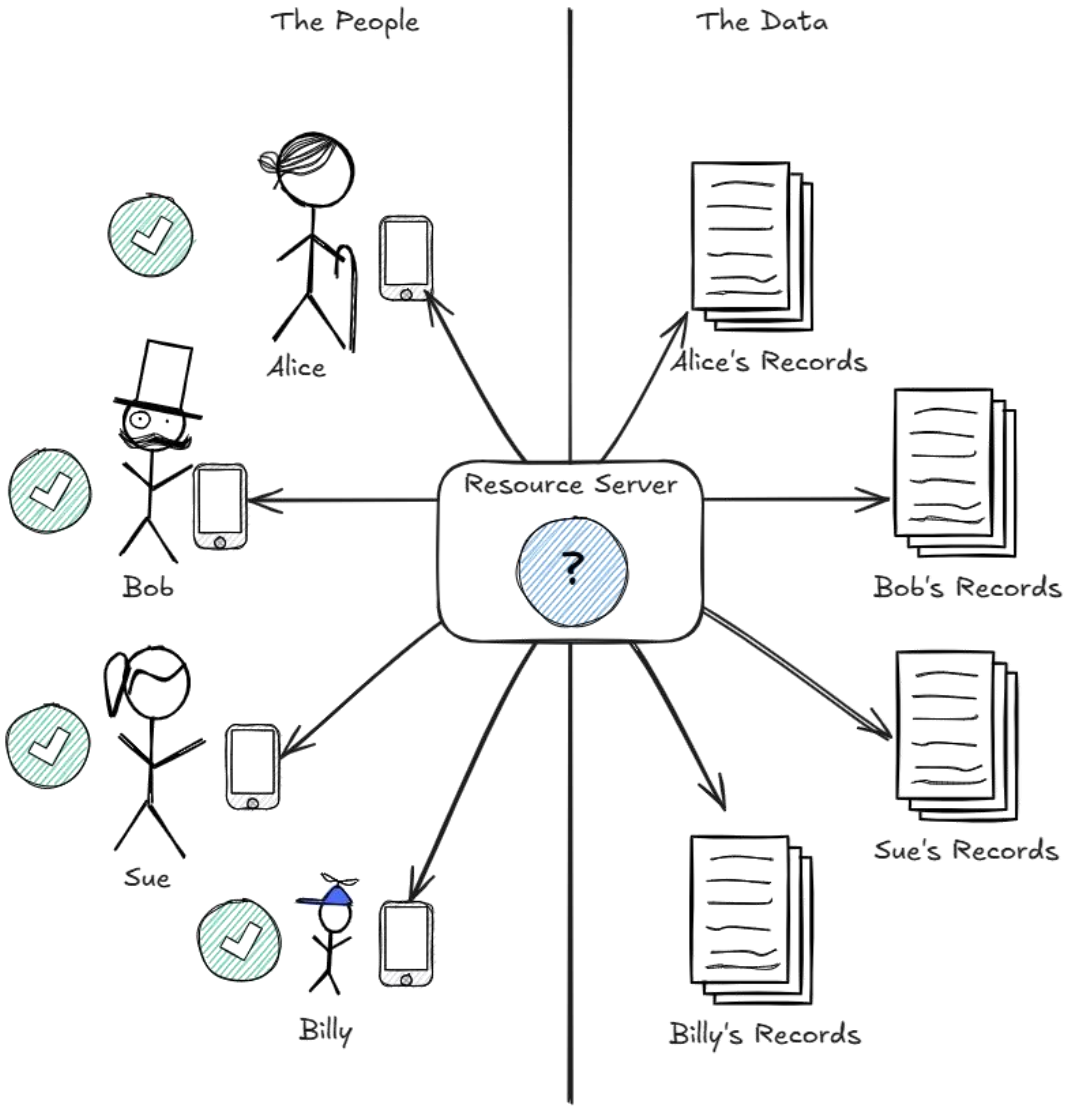

The People
The Data
Alice
Alice's Records
Bob
Bob's Records
Sue
Sue's Records
Billy
Billy's Records


The People
The Data
Alice
Alice's Records
Bob
Bob's Records
Sue
Sue's Records
Billy
Billy's Records

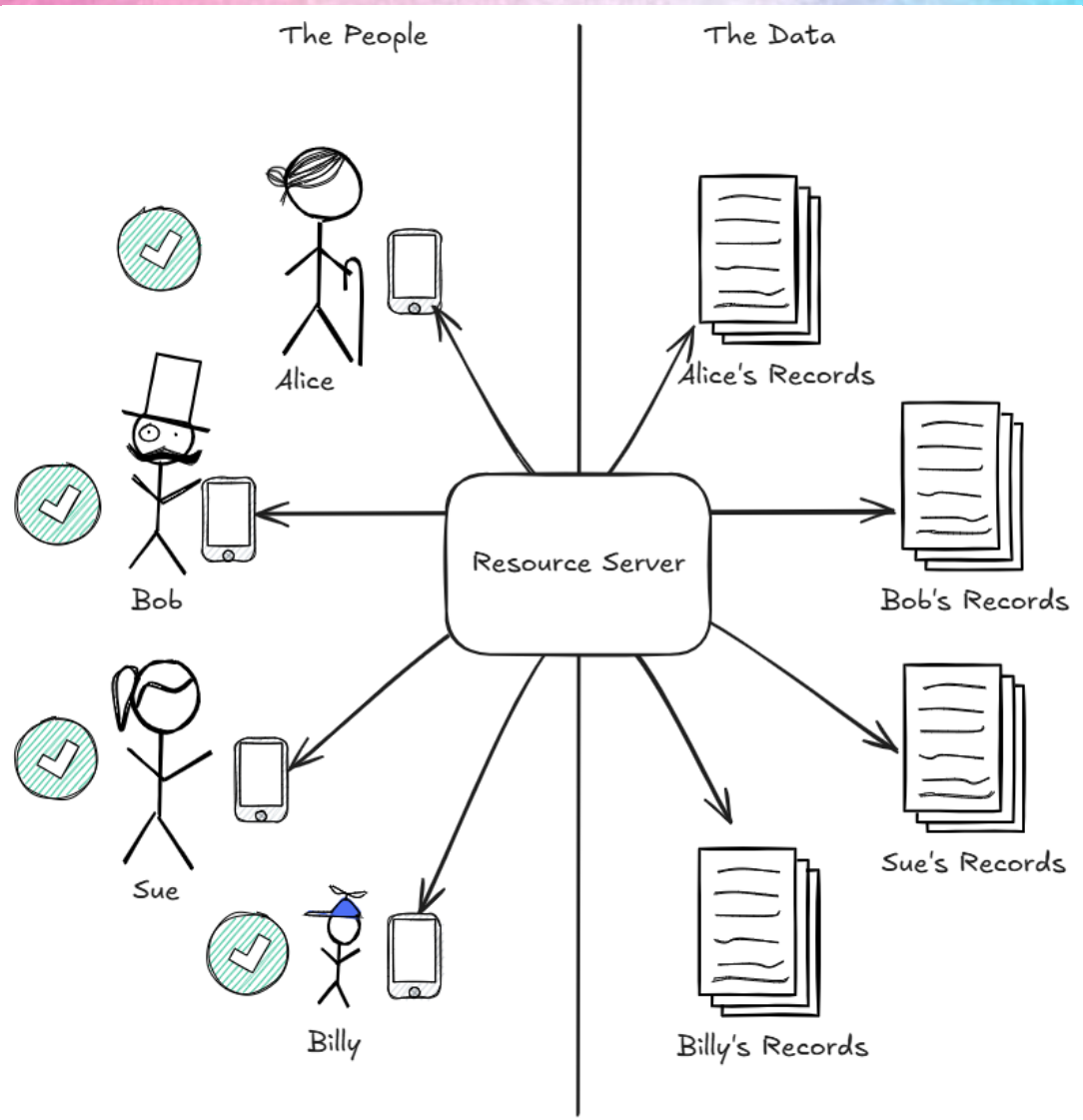

The People
The Data
Alice
Bob
Sue
Billy
Resource Server
Alice's Records
Bob's Records
Sue's Records
Billy's Records


The People
The Data
Alice
Alice's Records
Bob
Resource Server
Bob's Records
Sue
Sue's Records
Billy
Billy's Records

User declares identity

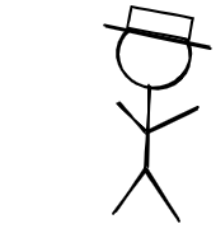


User declares identity
with username + password



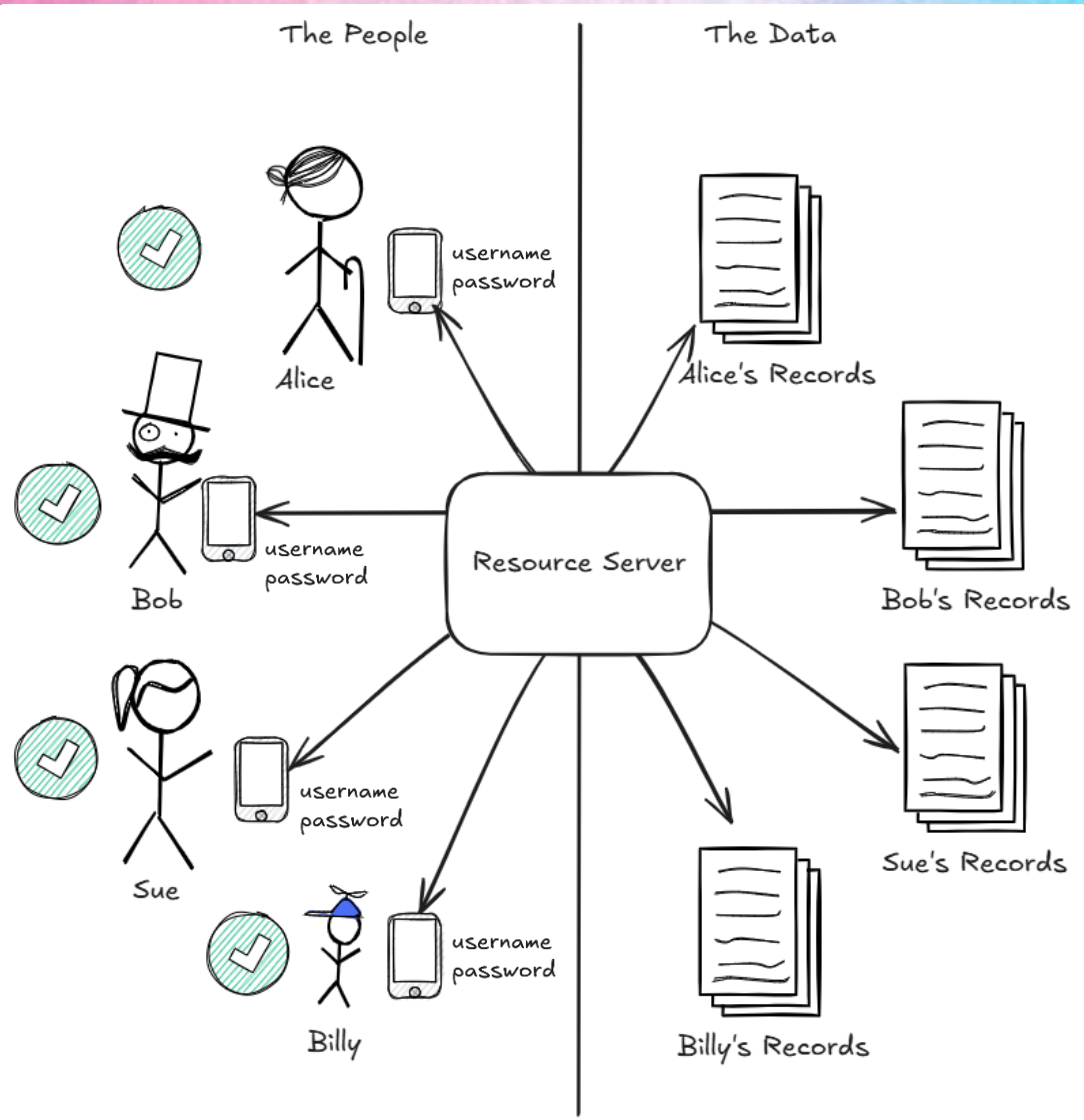
User declares identity
with username + password

Application stores username + password



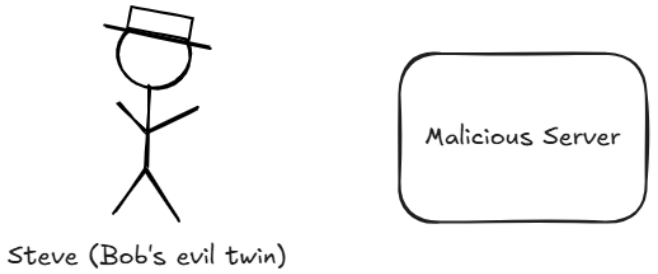
Steve (Bob's evil twin)

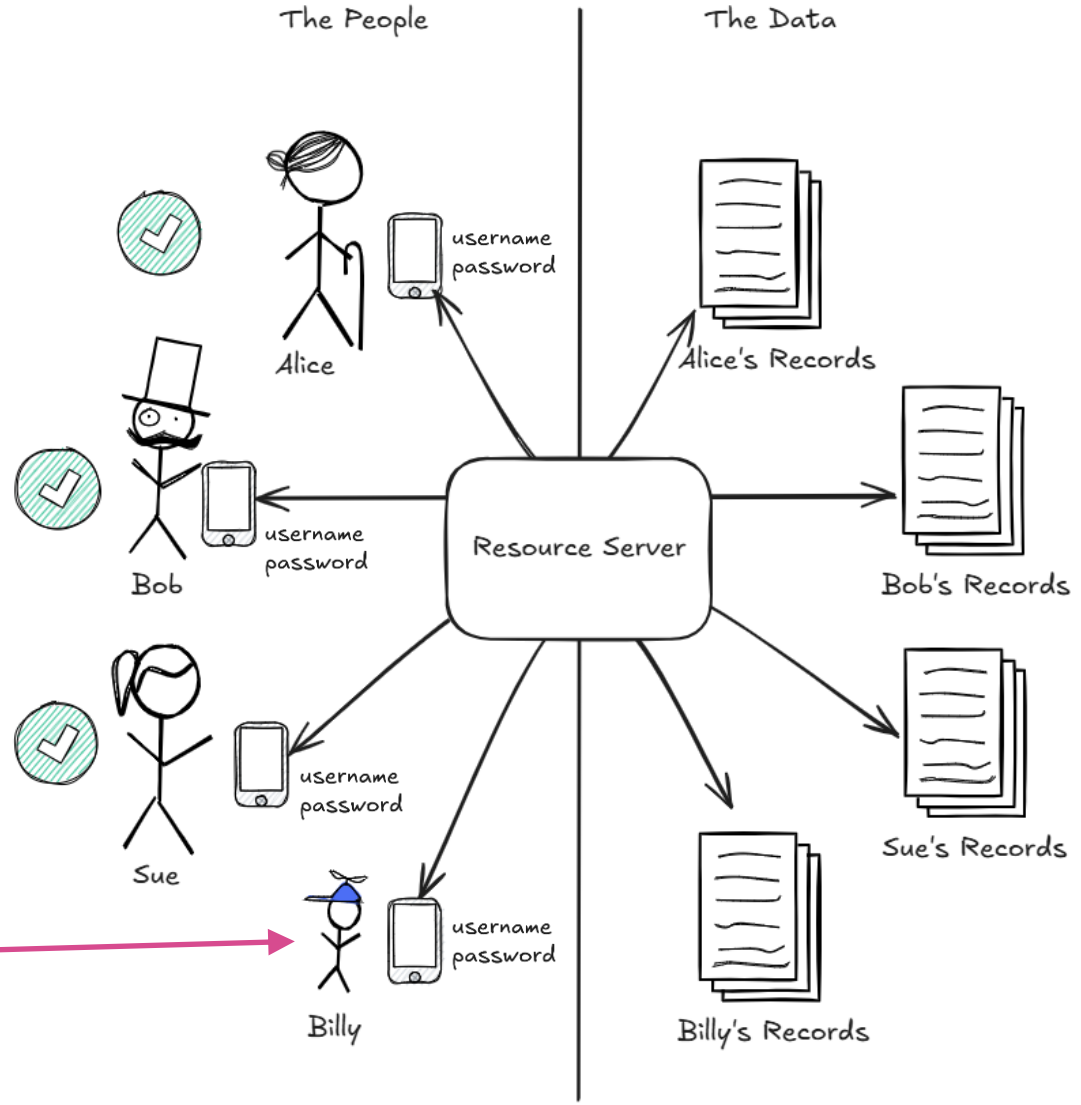




User declares identity
with username + password

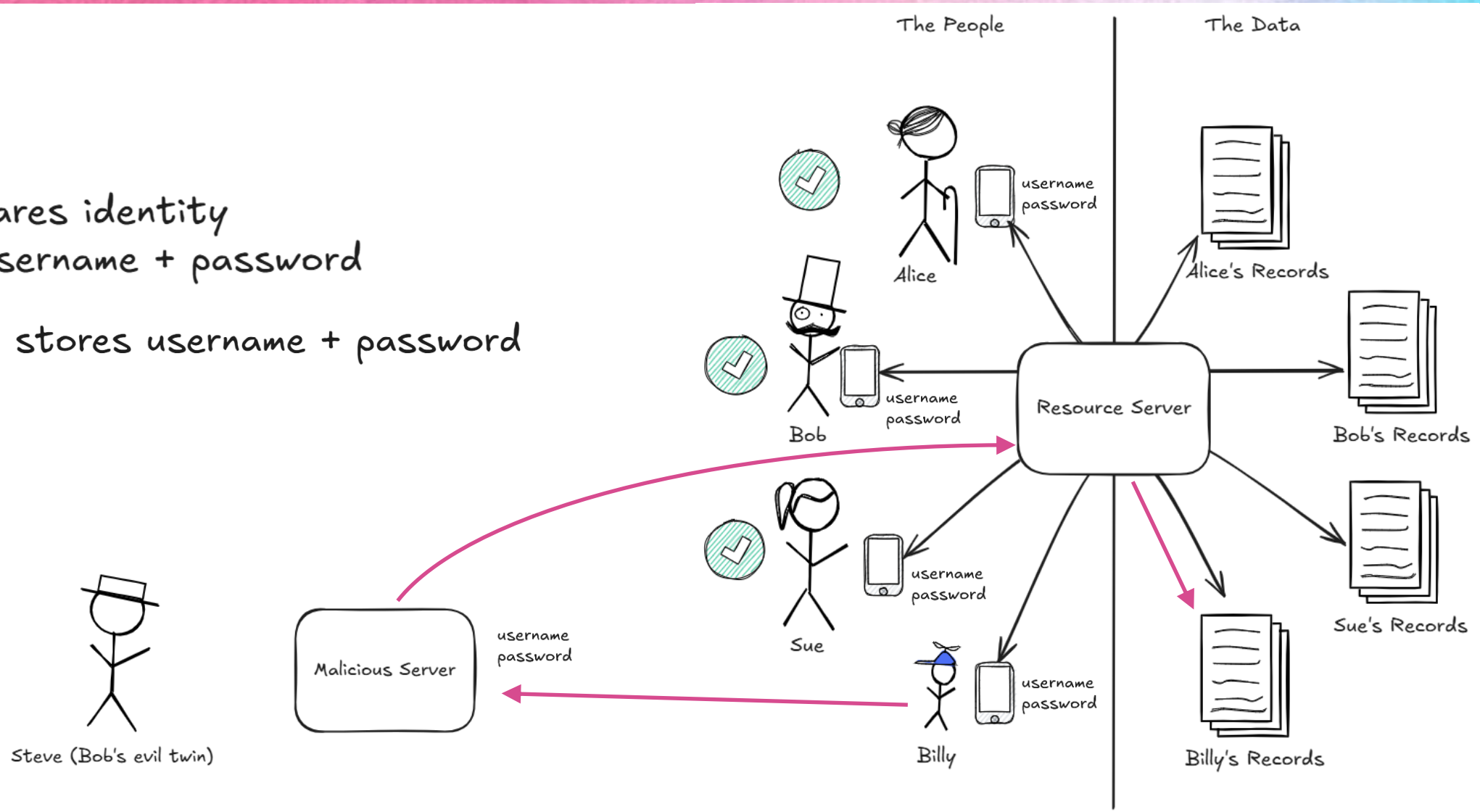
Application stores username + password





User declares identity
with username + password

Application stores username + password



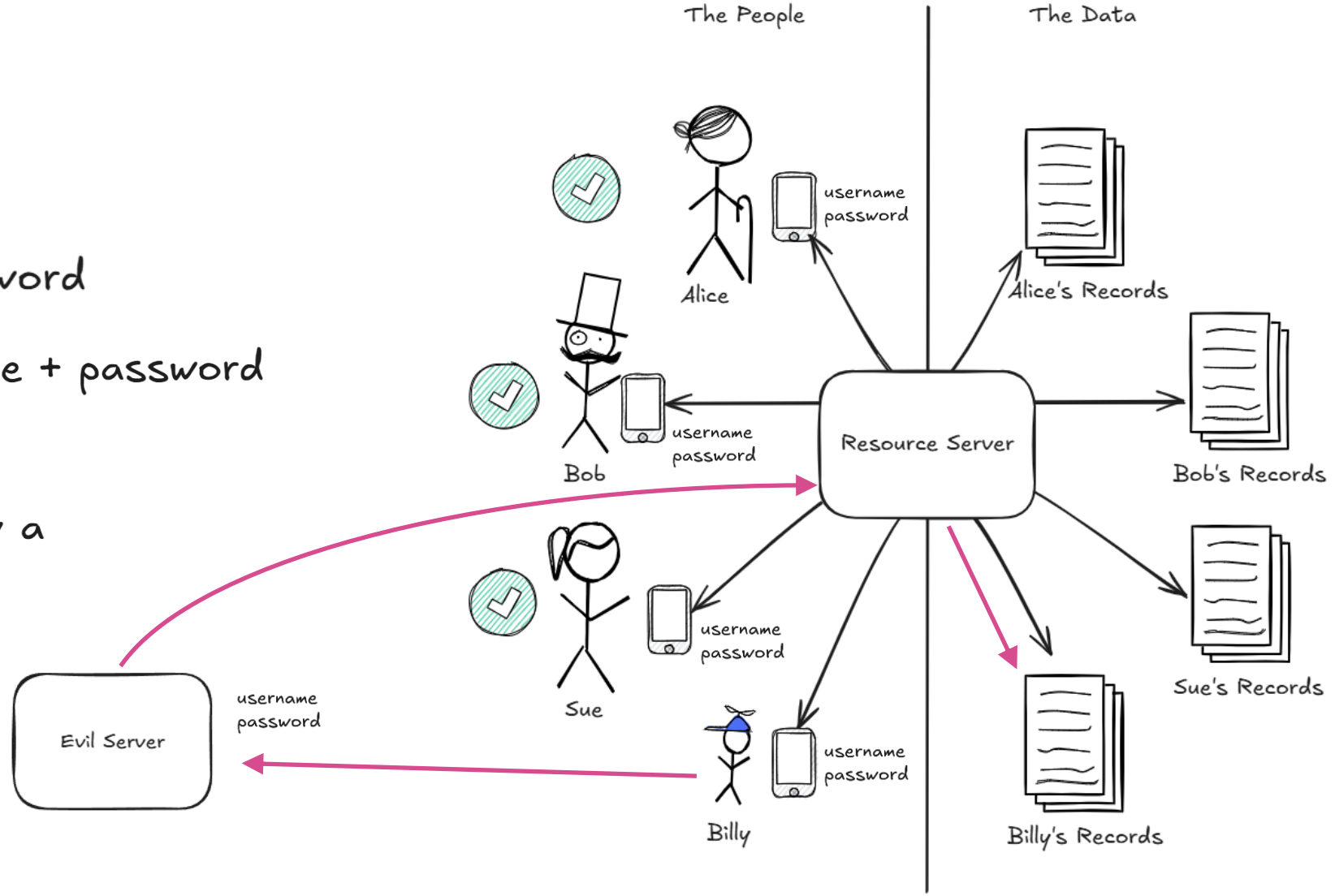
User declares identity
with username + password

Application stores username + password

Fundamental problem:
Application must supply a
user's credentials



Steve (Bob's evil twin)



What is OAuth 2.0?

What is OAuth 2.0?

> The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by **orchestrating an approval interaction between the resource owner and the HTTP service**, or by allowing the third-party application to obtain access on its own behalf.

RFC 6749

What is OAuth 2.0?



A framework for granting access to data

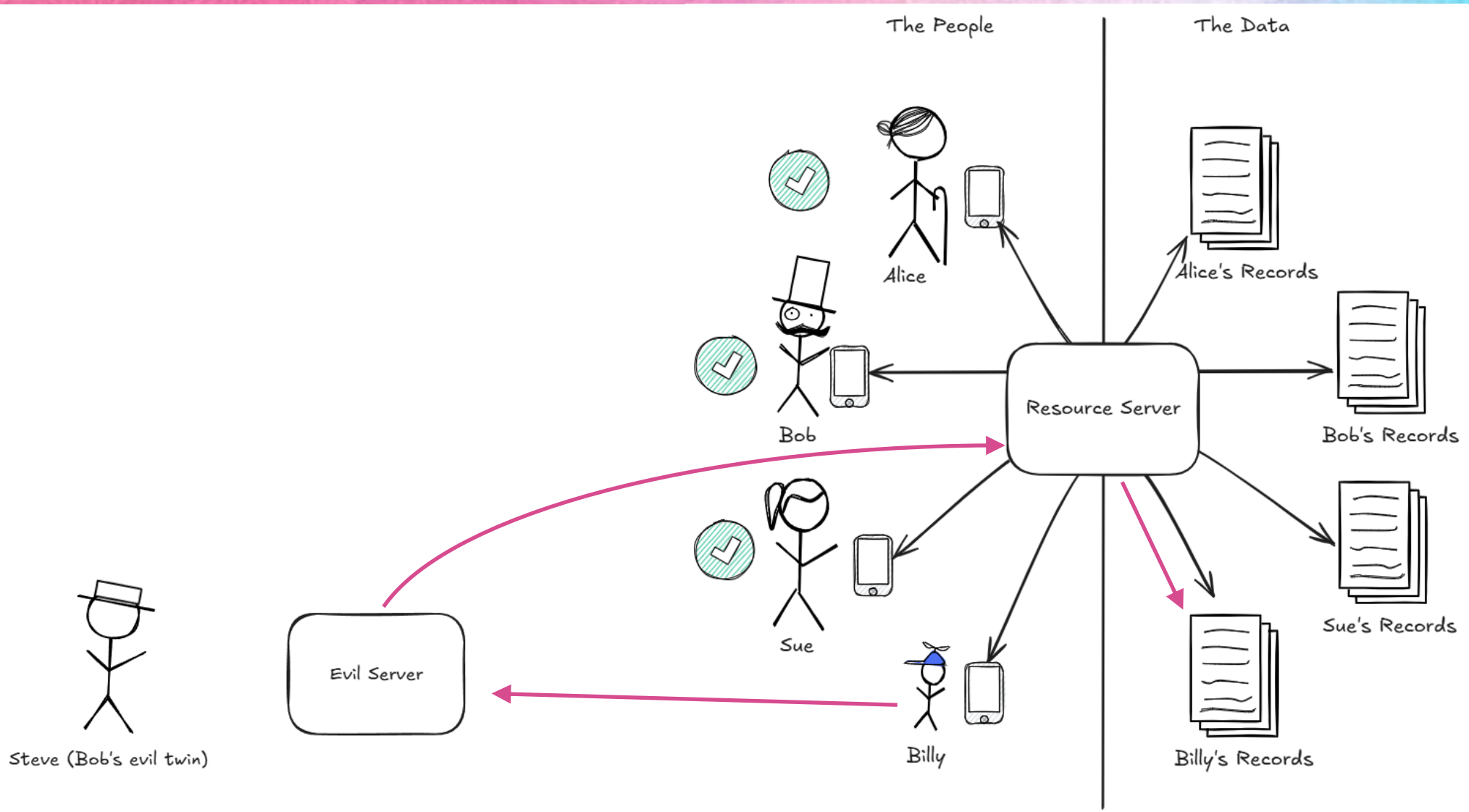


Data access is not tied to user credentials

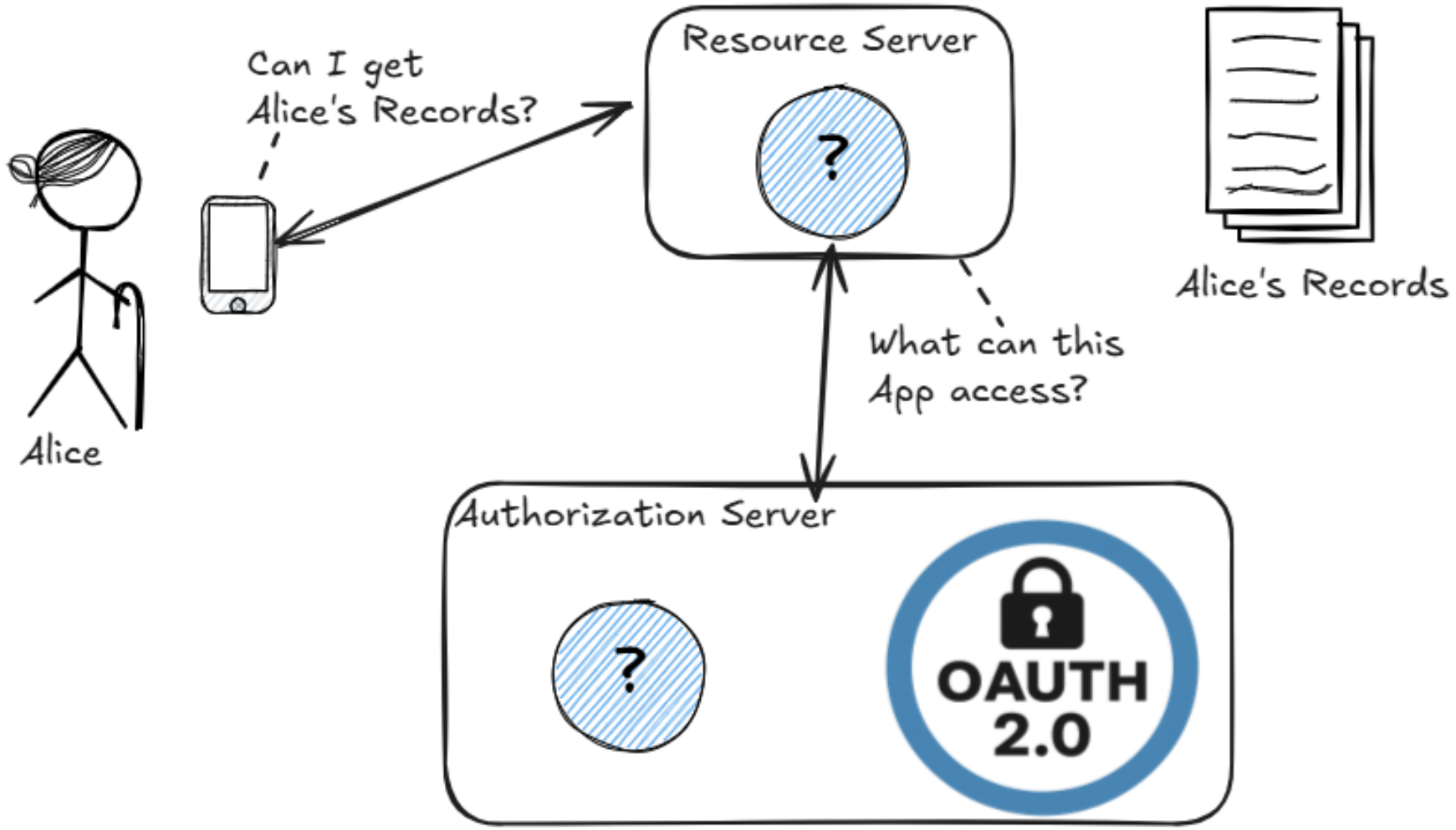


An extensible protocol

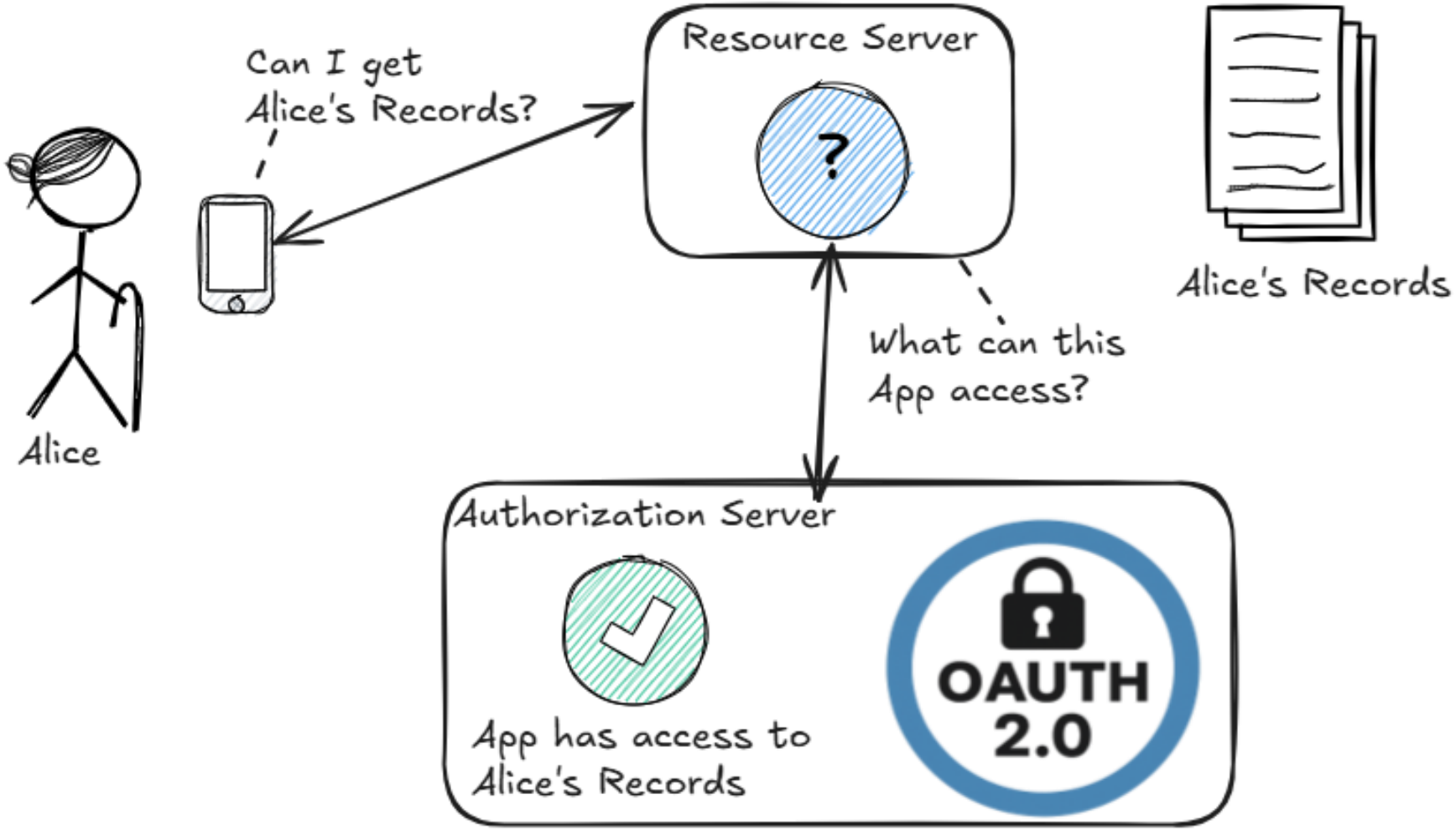
How Does OAuth 2.0 Avoid This?



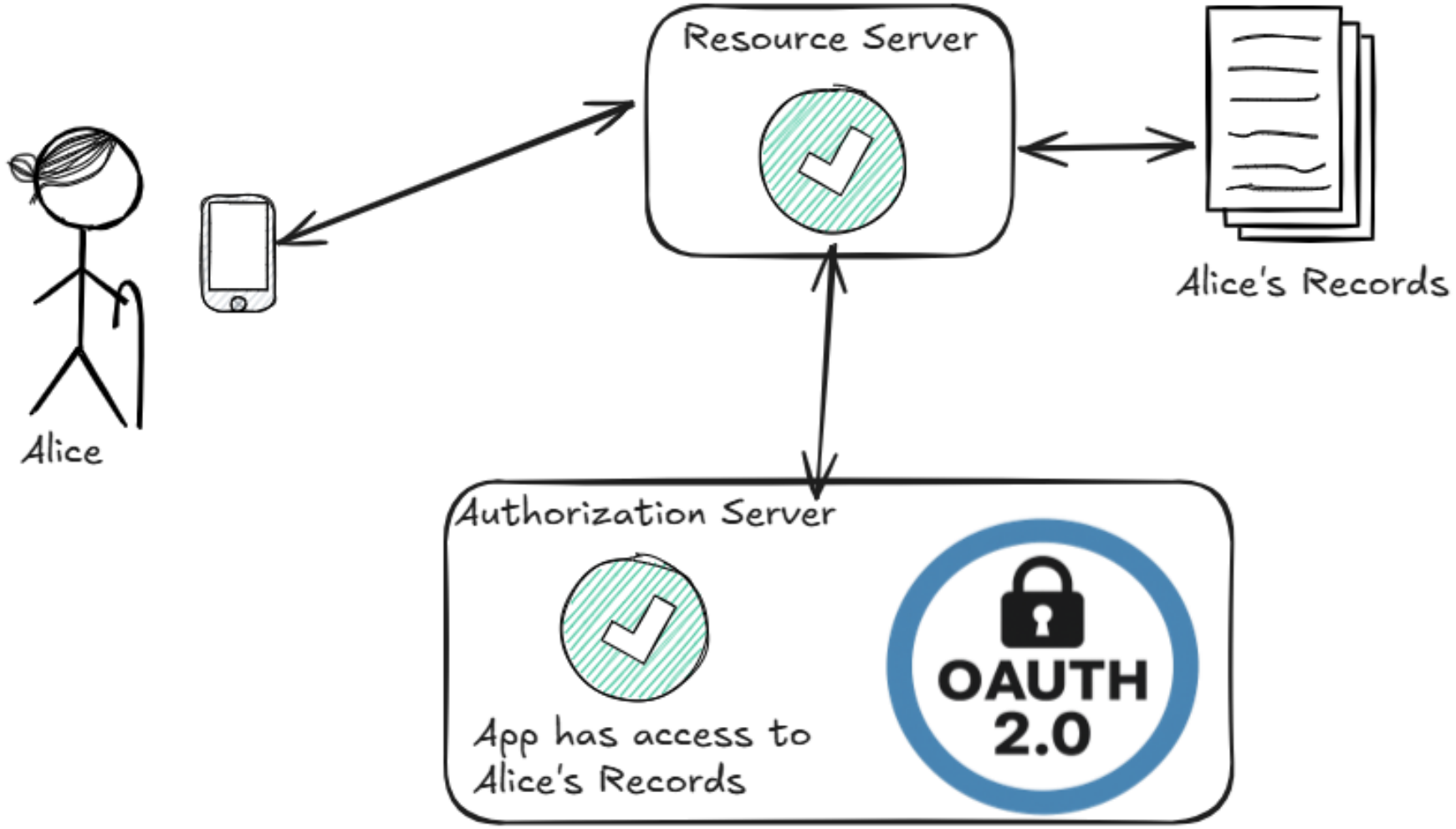
A Resource Server AND An Authorization Server



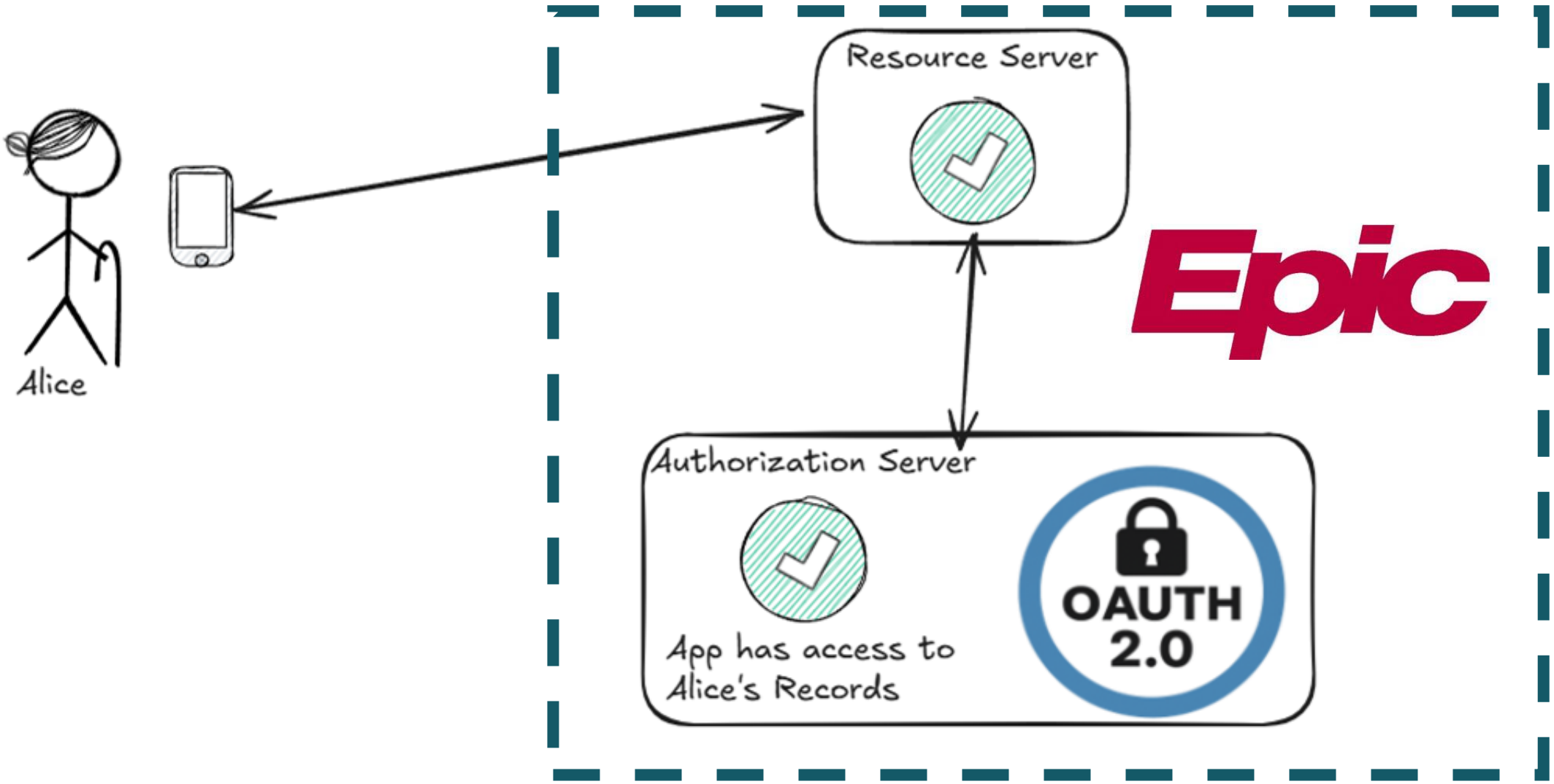
A Resource Server AND An Authorization Server



A Resource Server AND An Authorization Server

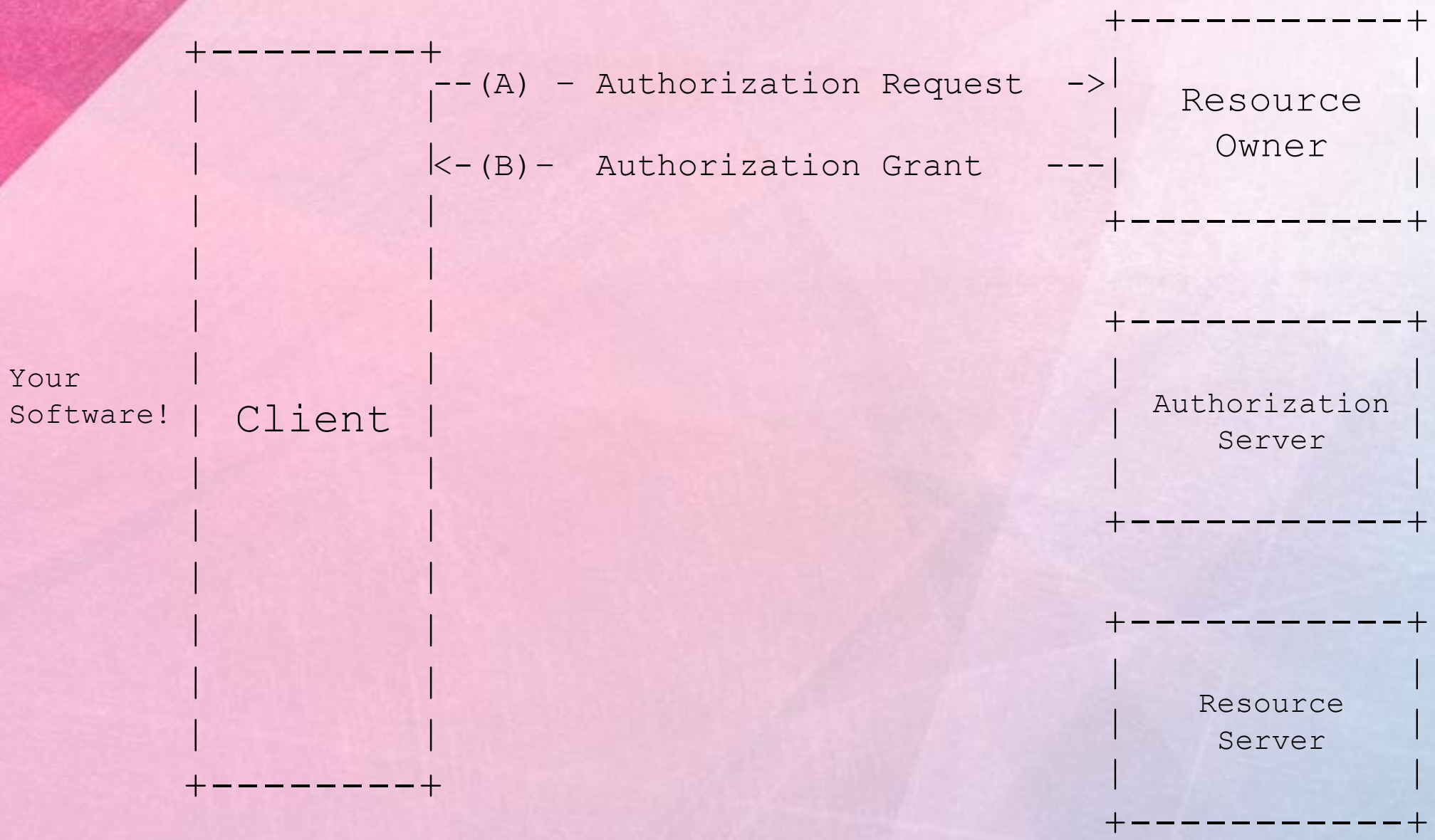


A Resource Server AND An Authorization Server



The Call Flow & Grant Types

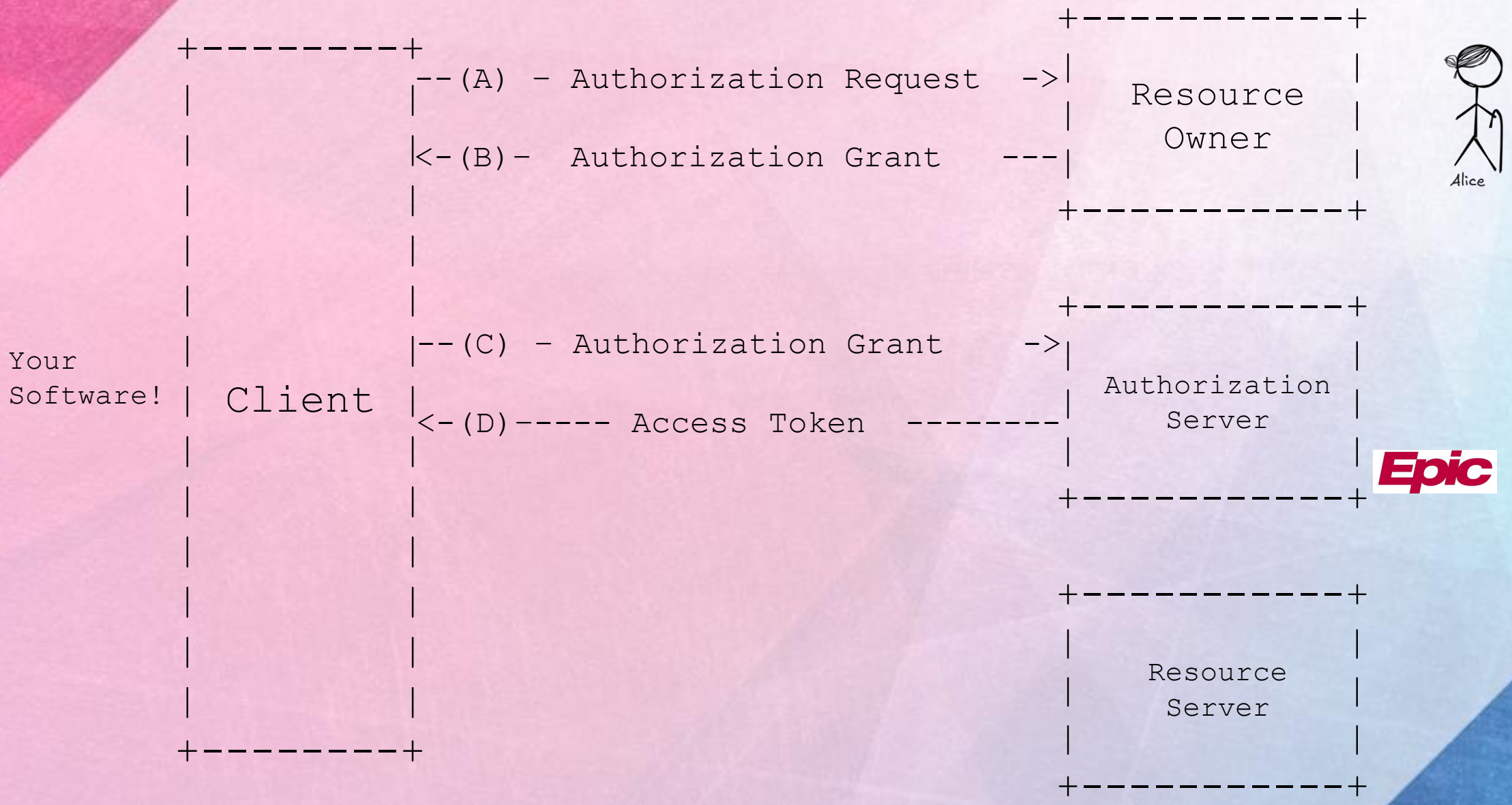
A Series of Calls, in Stages





RFC 6749

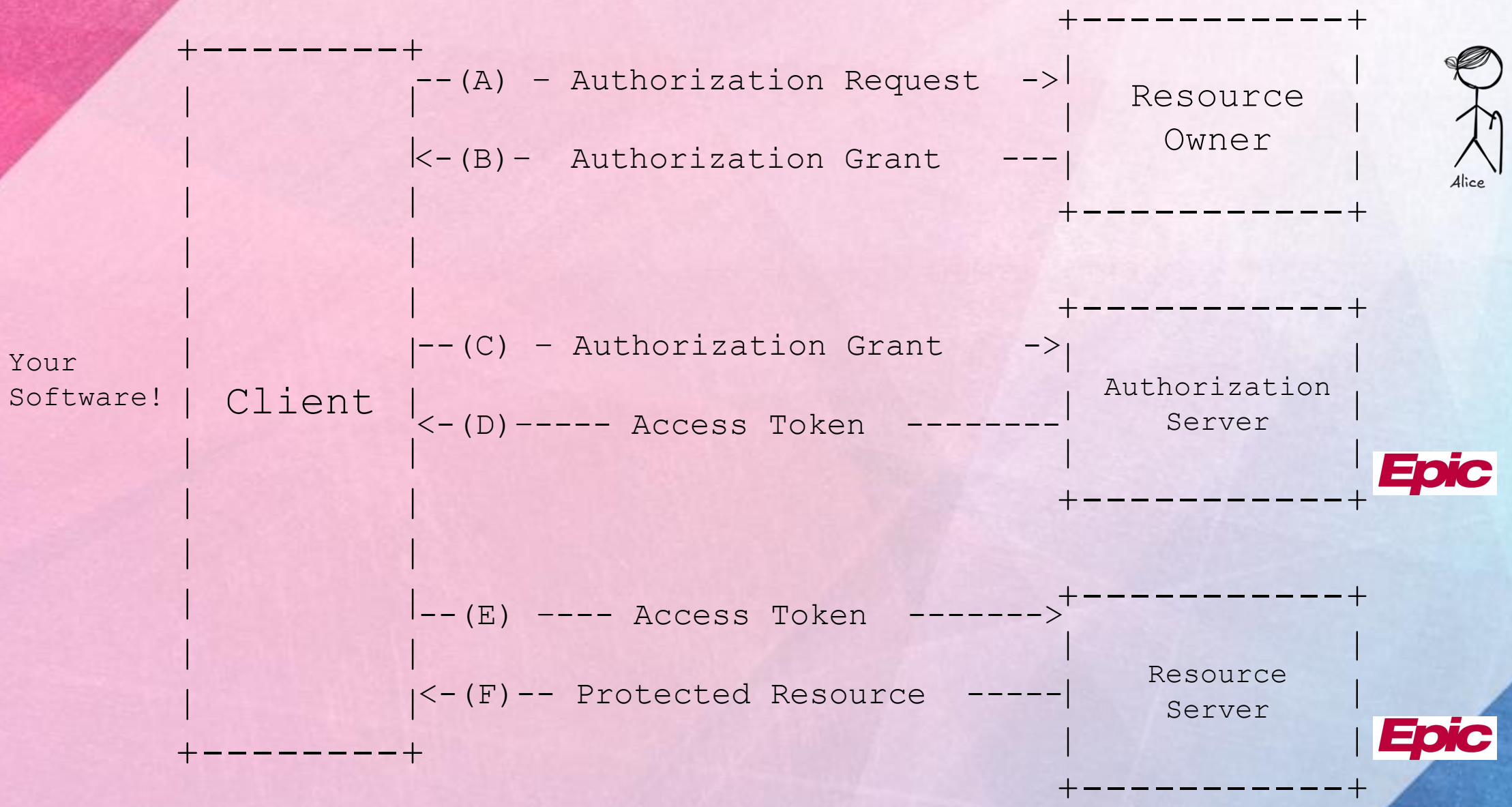
A Series of Calls, in Stages



Epic

RFC 6749

A Series of Calls, in Stages

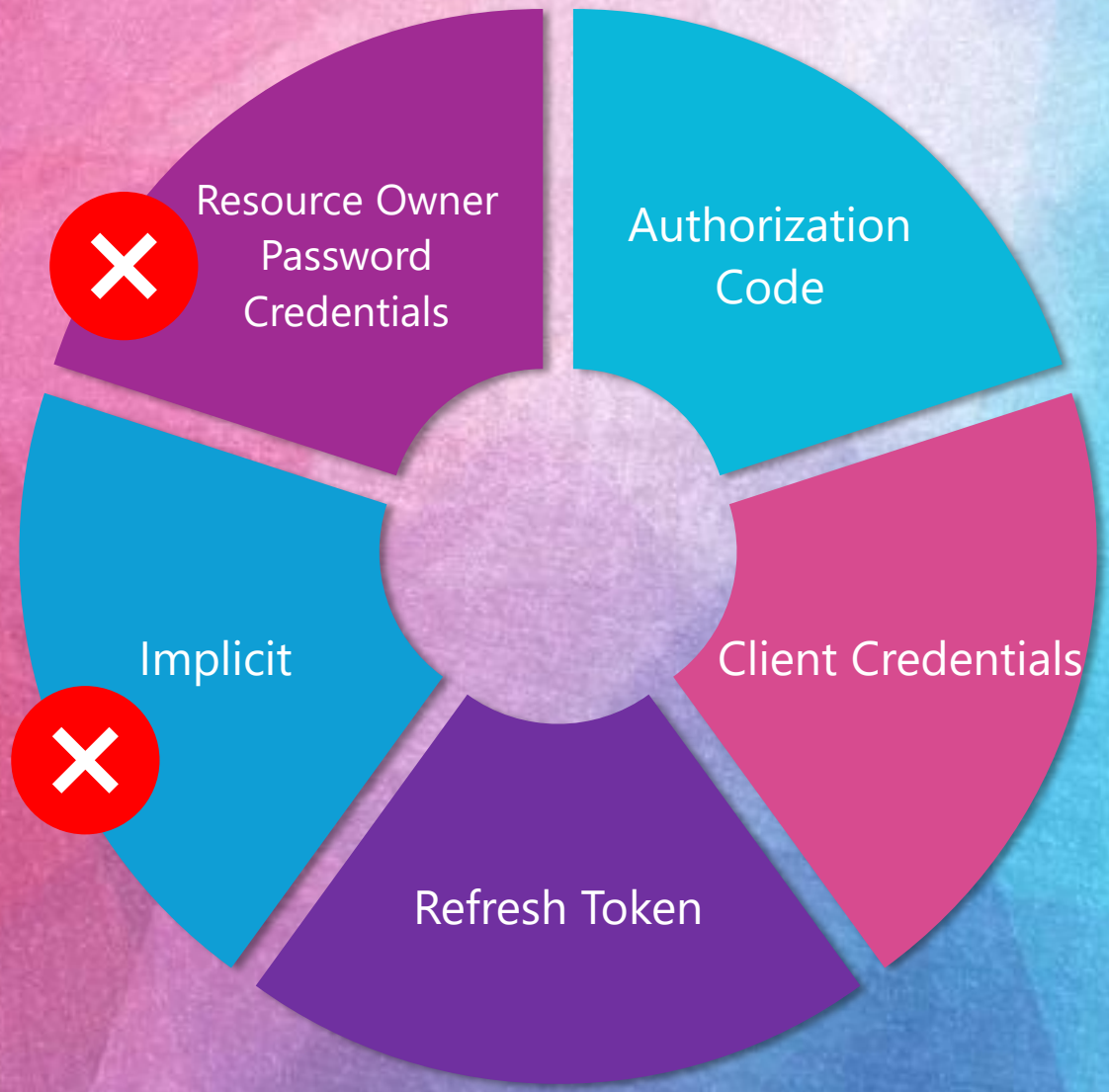


Epic

Epic

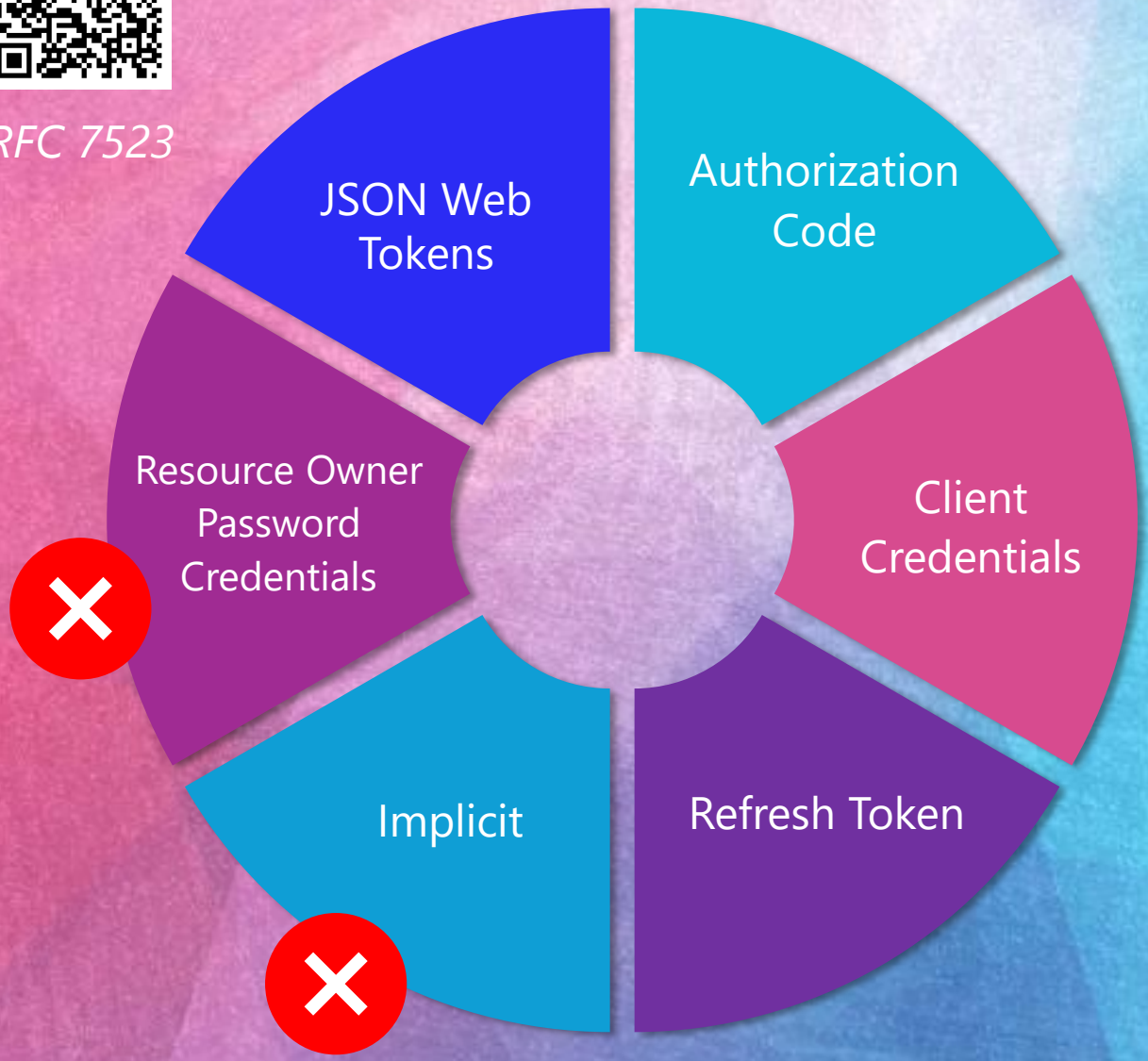
RFC 6749

Grant Types

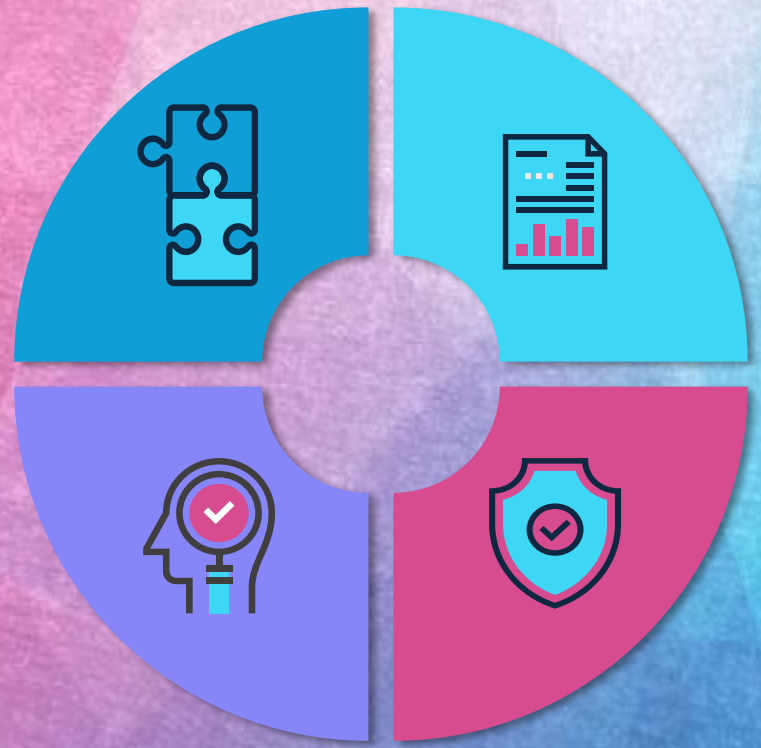

Resource Owner Password Credentials
Authorization Code
Implicit
Client Credentials
Refresh Token

Grant Types

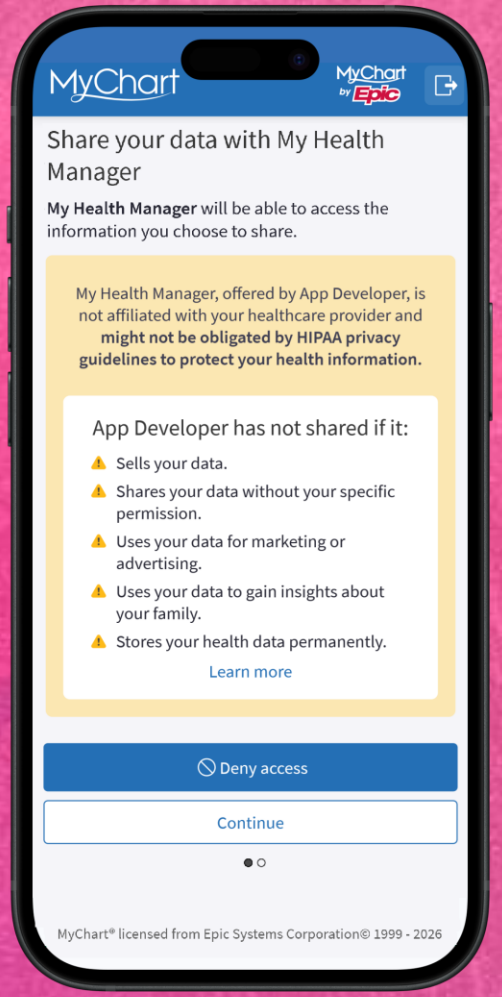
RFC 7523

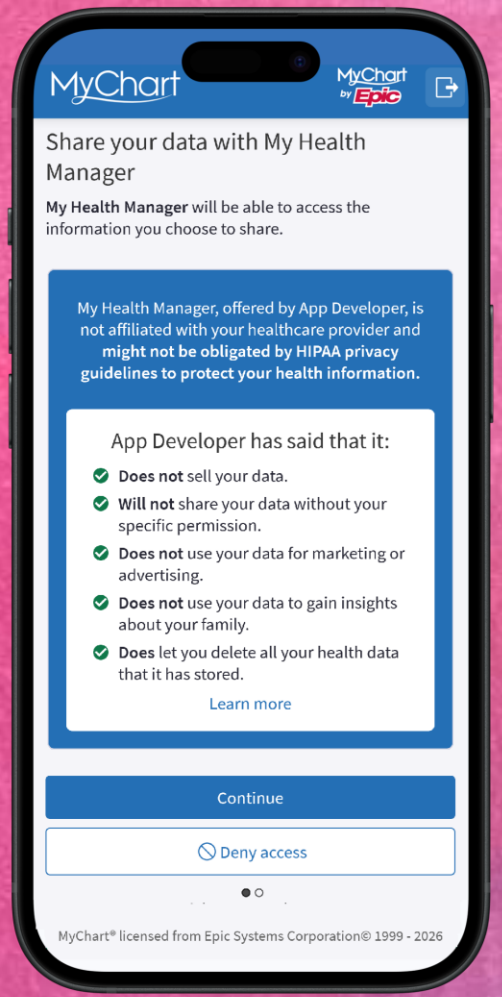


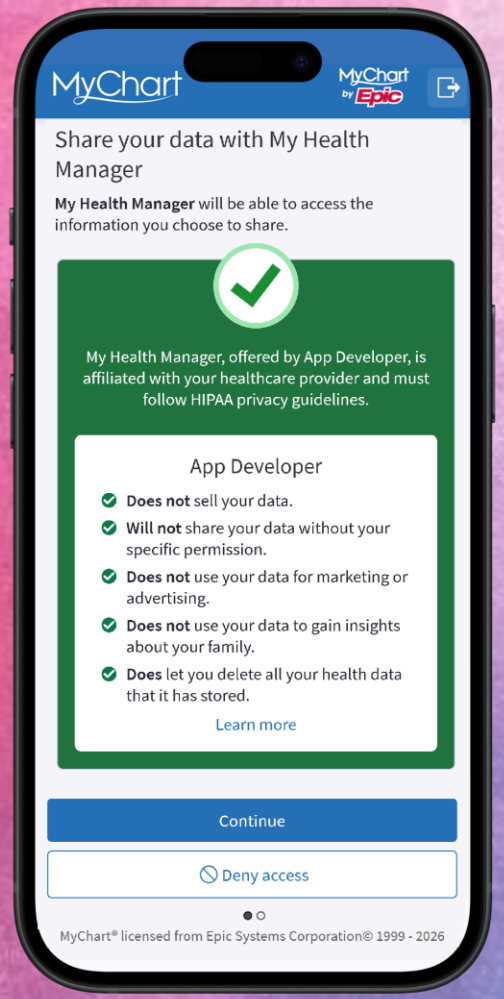
Unpacking Each Piece



Authorizing Patient Access



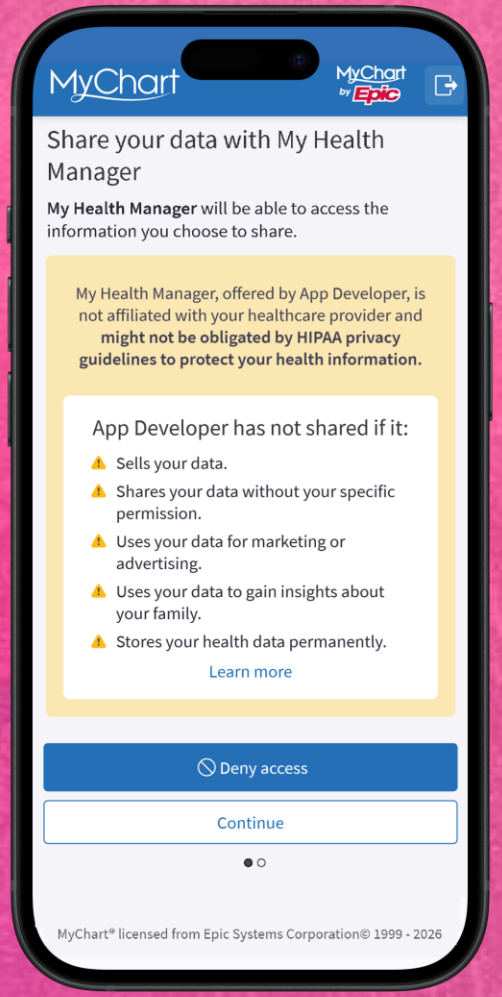


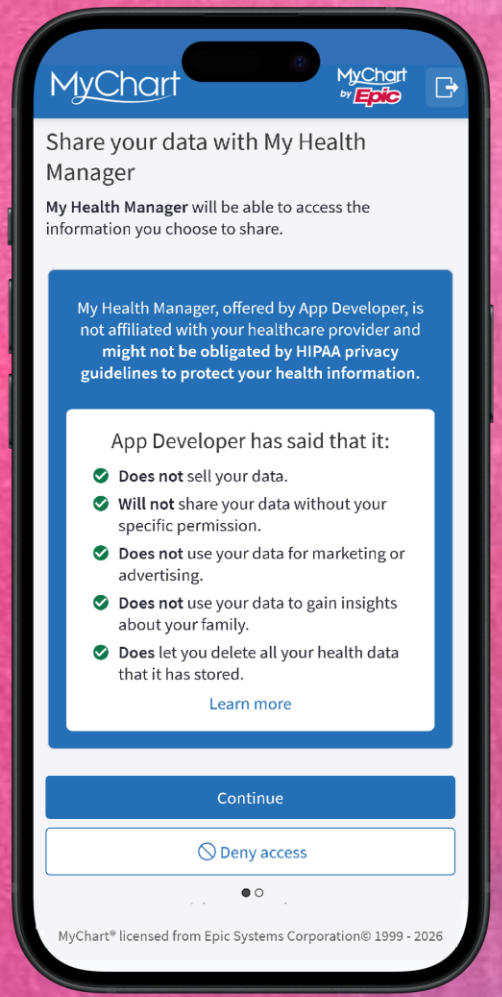


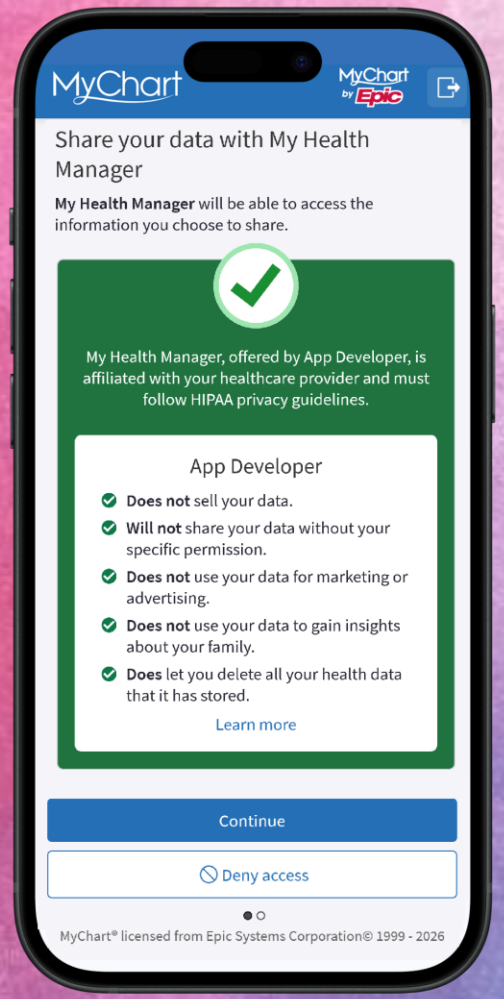


Design Strategies to Increase Patient Data Sharing

Authorizing Patient Access



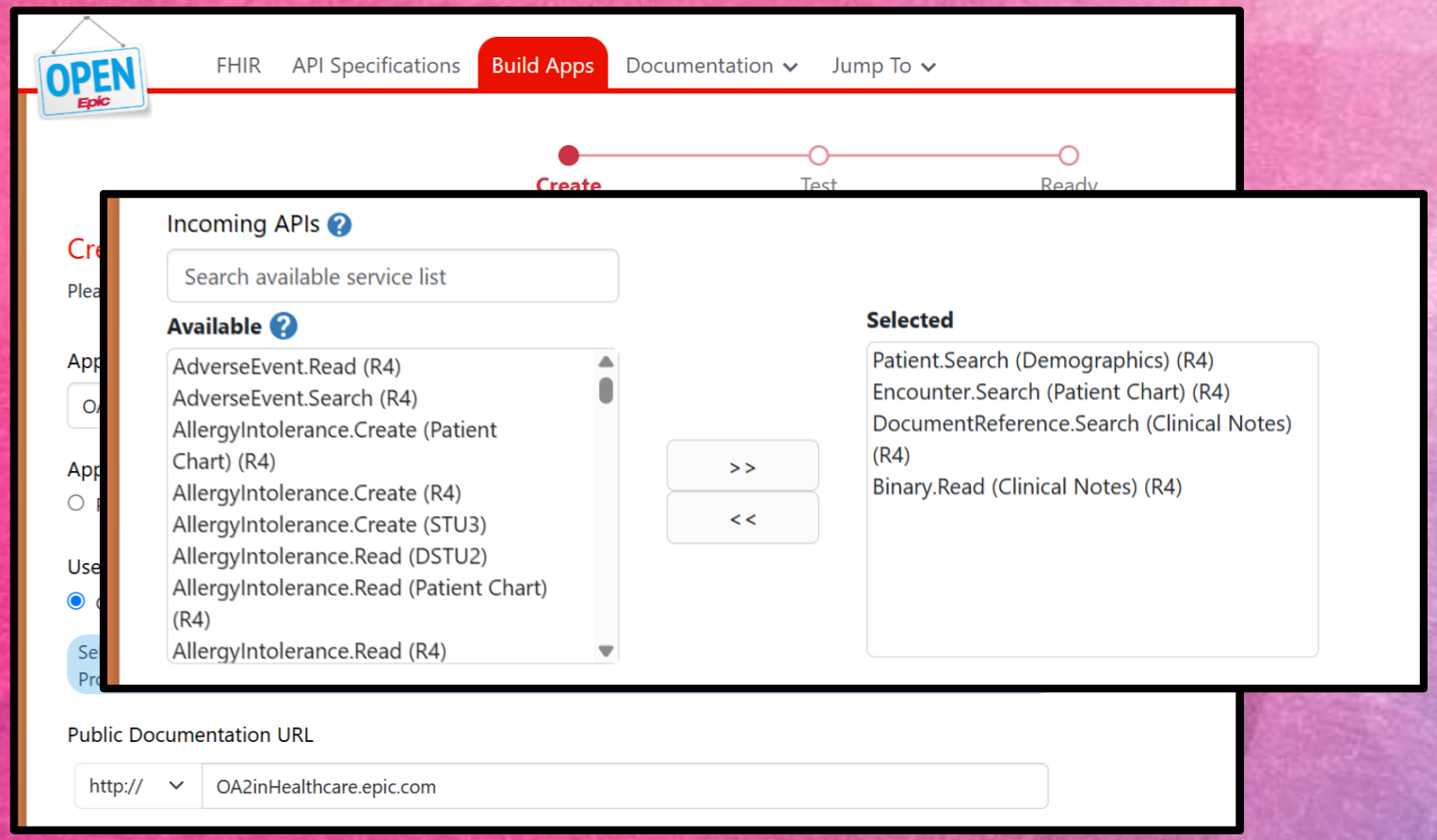






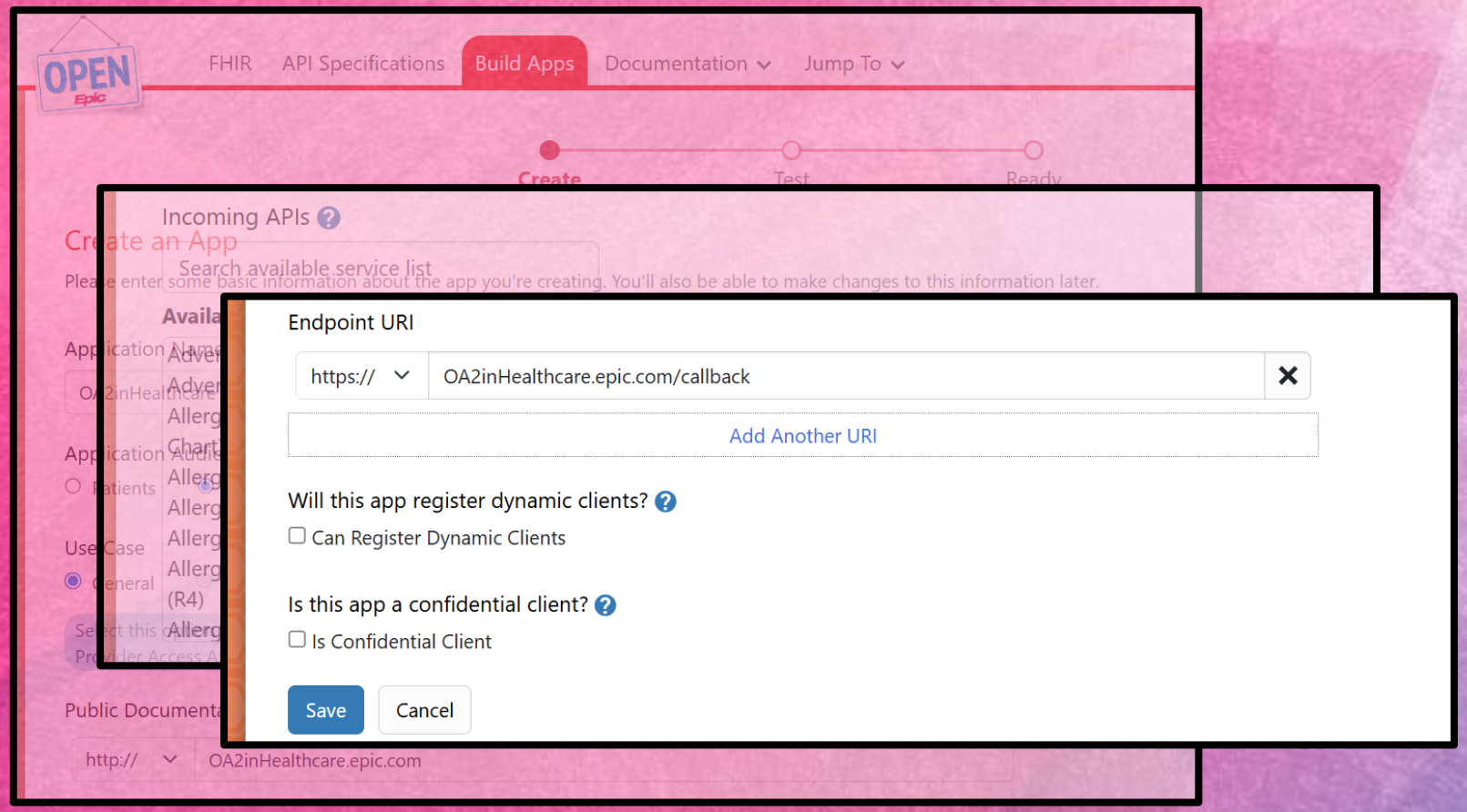
Design Strategies to Increase Patient Data Sharing

Authorizing Client Access



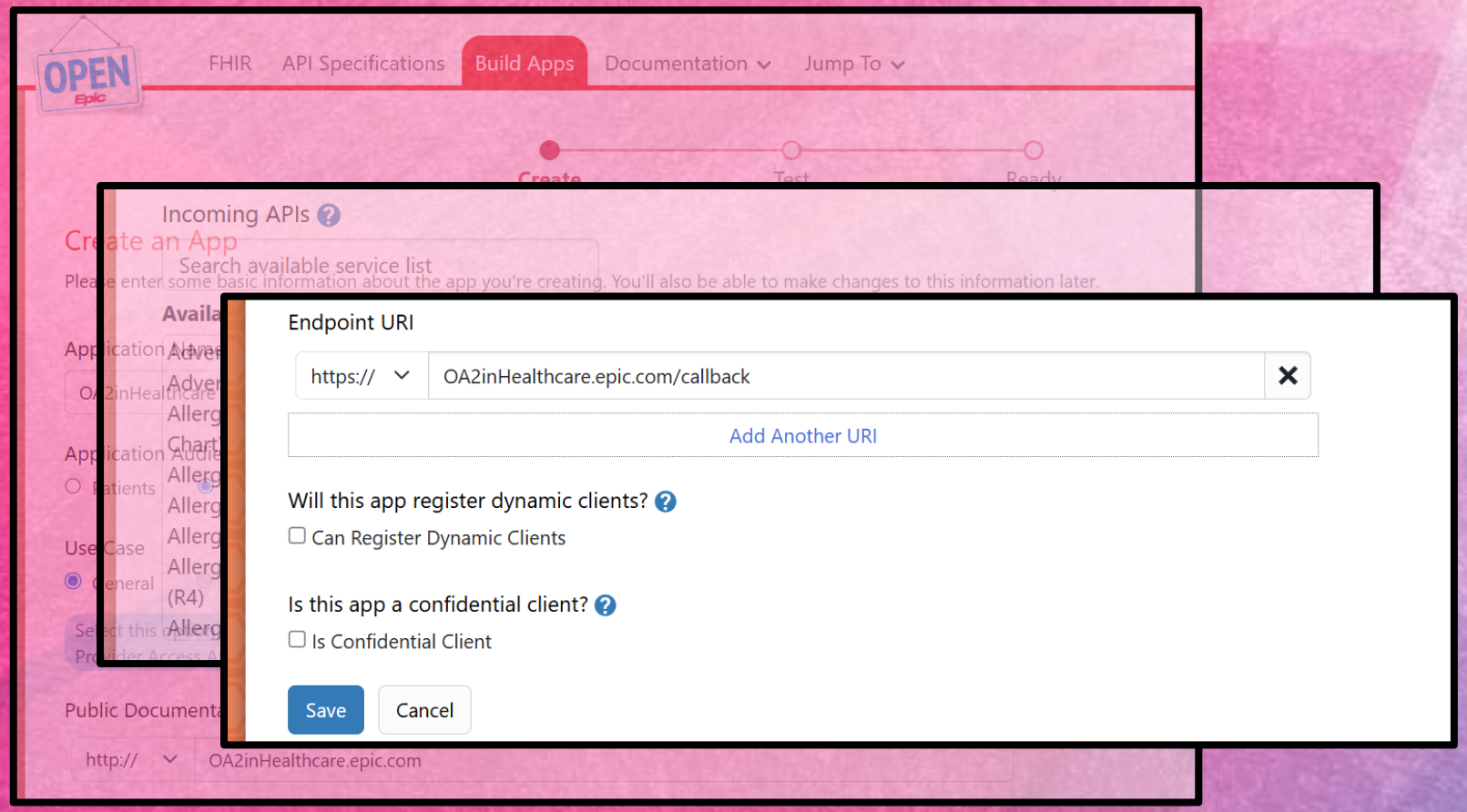


Authorizing Client Access





Authorizing Client Access

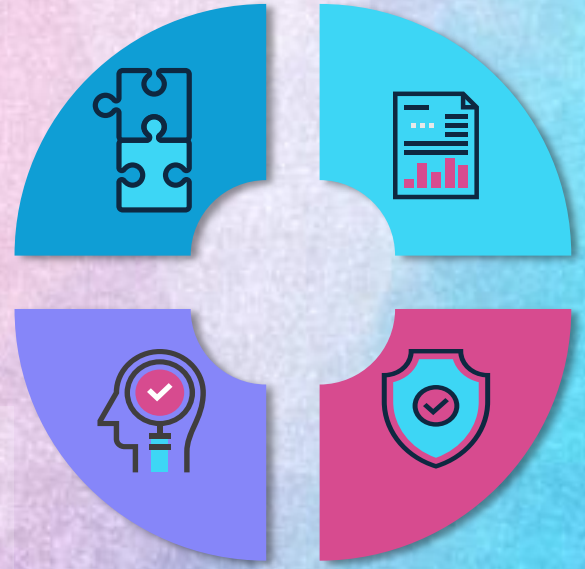




OEC19 Navigating App Governance A Provider's Perspective

Wednesday, 10:30 AM

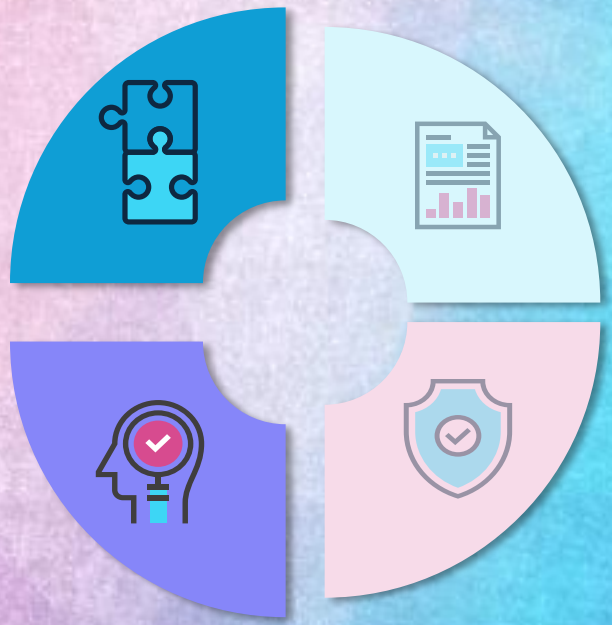
Authorization Request Scope String



```
https://fhir.epic.com/interconnect-fhir-oauth/oauth2/authorize?
  client_id=e3073934-68c1-4ca7-9b59-3d8b934187f1&
  scope=launch%20openid%20fhirUser&
  ...
```

Authorization Request OIDC

SSO  + Account Creation 



```
scope=launch%20openid%20fhirUser&
   ...
```

Identifying the User and On-the-Fly Account Creation

Authorization Request OIDC

SSO  + Account Creation 

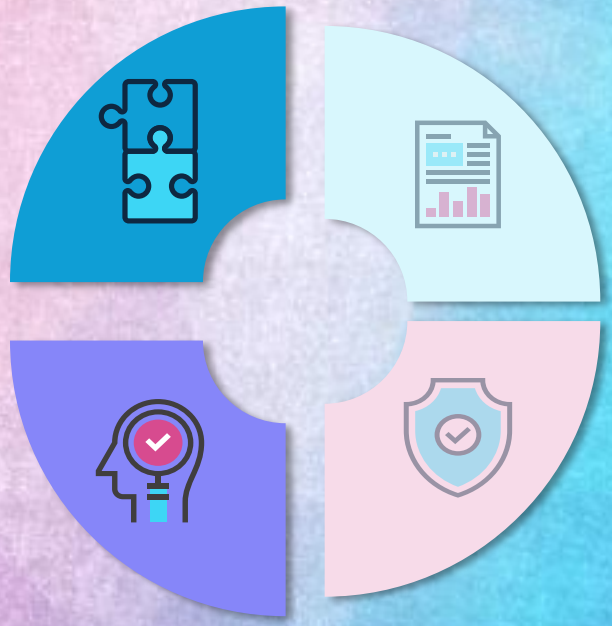




```
scope=launch%20openid%20fhirUser&
   ...
&nonce=5xXdlZSeqrAB5aH6cOyPy5zs
```

Identifying the User and On-the-Fly Account Creation

Authorization Request OIDC



scope=openid fhirUser ...

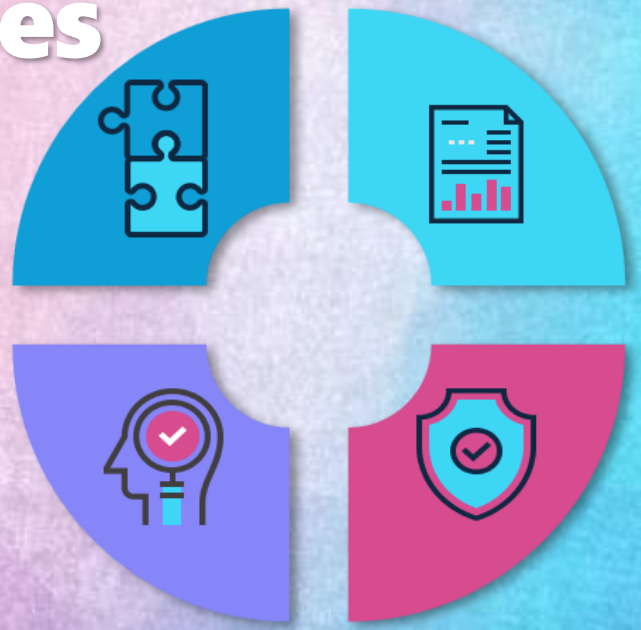
`fhirUser`	Absolute reference to the FHIR resource representing the user launching the app. See the HL7 documentation for more details.	The *fhirUser* claim will only be present if app has the R4 (or greater) SMART on FHIR version selected, and requests both the *openid* and *fhirUser* scopes in the request to the authorize endpoint. See the remark for the `sub` claim above for more information about what the resource returned in these claims represents.

scope=openid profile ...

`preferred_username`	The user's LDAP/AD down-level logon name.	This claim is supported in Epic version February 2021 and later. The *preferred_username* claim will be present only if the app requests both the *openid* and *profile* scopes in the request to the authorize endpoint.

OpenID Connect
id_tokens

Authorization Request Launch Scopes



scope=launch/patient ...

Search for a Patient

Patient | Patient Encounter

Name/MRN | Birth Date | Legal Sex

SSN | Zip Code | Phone #

scope=launch/encounter ...

Patient Encounter Selection

Patient
Mychart, Theodore [202500]

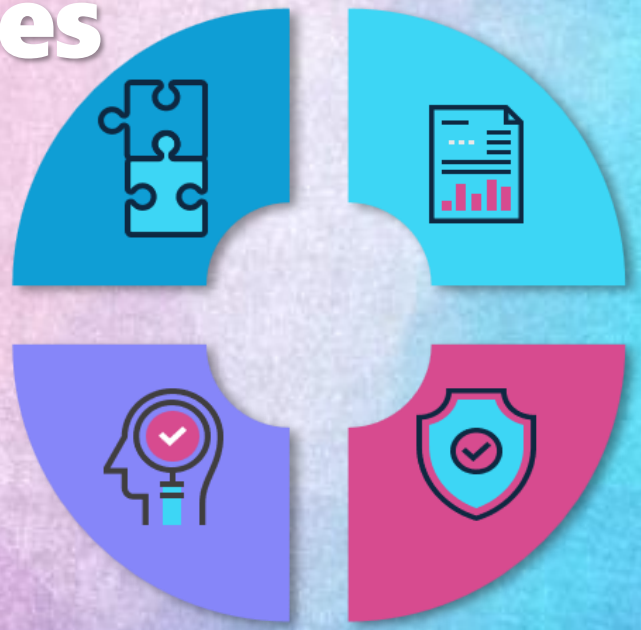
Patient Encounter

+ Create an Encounter | Show Filters

Contact Date	Contact Type	Provider	Department	Status
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Postpartum Visit	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Education	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Orders Only	Physician Family Medicine, MD	EMC FAMILY MEDICINE	
05/21/2019	Appointment	Physician Family Medicine, MD	INITIAL DEPT	No Show
11/08/2013	Admission	Cardiologist Cupid, MD	EMH CARDIAC ICU	Pend Adm
11/06/2013	Office Visit	Physician Family Medicine, MD	EMC FAMILY MEDICINE	Completed
11/29/2012	Patient Message	Physician Family Medicine, MD	EMC FAMILY MEDICINE	

SMART – Standalone apps

Authorization Request Launch Scopes



scope=launch/patient ...

Search for a Patient

Patient | Patient Encounter

Name/MRN | Birth Date | Legal Sex

SSN | Zip Code | Phone #

scope=launch/encounter ...

Patient Encounter Selection

Patient
Mychart, Theodore [202500]

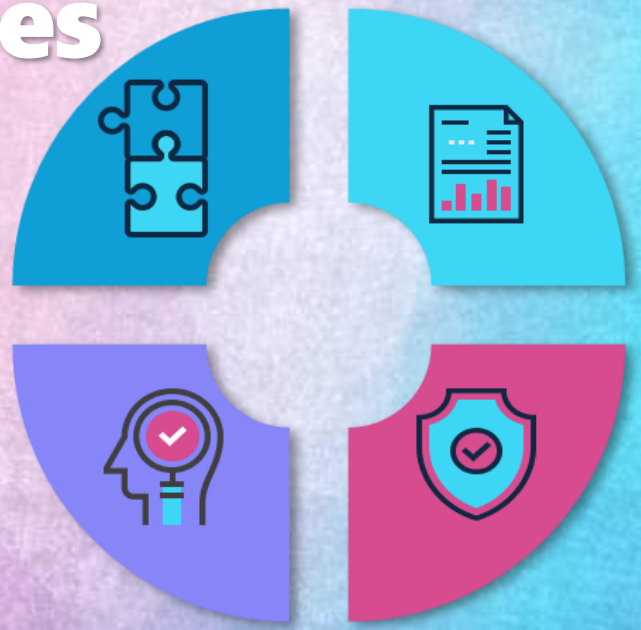
Patient Encounter

+ Create an Encounter | Show Filters

Contact Date	Contact Type	Provider	Department	Status
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Postpartum Visit	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Education	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Orders Only	Physician Family Medicine, MD	EMC FAMILY MEDICINE	
05/21/2019	Appointment	Physician Family Medicine, MD	INITIAL DEPT	No Show
11/08/2013	Admission	Cardiologist Cupid, MD	EMH CARDIAC ICU	Pend Adm
11/06/2013	Office Visit	Physician Family Medicine, MD	EMC FAMILY MEDICINE	Completed
11/29/2012	Patient Message	Physician Family Medicine, MD	EMC FAMILY MEDICINE	

SMART – Standalone apps

Authorization Request Launch Scopes



scope=launch/patient ...

Search for a Patient

Patient | Patient Encounter

Name/MRN | Birth Date | Legal Sex

SSN | Zip Code | Phone #

scope=launch/encounter ...

Patient Encounter Selection

Patient
Mychart, Theodore [202500]

Patient Encounter

+ Create an Encounter | Show Filters

Contact Date	Contact Type	Provider	Department	Status
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Patient Self-Triage	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Postpartum Visit	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Education	Generic Physician System Default, MD	EMC FAMILY MEDICINE	
05/24/2019	Orders Only	Physician Family Medicine, MD	EMC FAMILY MEDICINE	
05/21/2019	Appointment	Physician Family Medicine, MD	INITIAL DEPT	No Show
11/08/2013	Admission	Cardiologist Cupid, MD	EMH CARDIAC ICU	Pend Adm
11/06/2013	Office Visit	Physician Family Medicine, MD	EMC FAMILY MEDICINE	Completed
11/29/2012	Patient Message	Physician Family Medicine, MD	EMC FAMILY MEDICINE	

SMART – Standalone apps

EHR Launch

Token Library

Health system analysts can configure an integration record to launch a URL, and when doing so, they can use a library of "tokens," which are substituted with identifiers from the user's context upon the launch of your application. The context column notes the minimum amount of information that must be available to the app at the time of launch to ensure data is populated.

In an OAuth2 SMART on FHIR workflow, these "tokens" are sent as separate key-value pairs, POSTed in the same response with (but not included in) the access token. Refer to Step 5: Your Application Exchanges the Authorization Code for an Access Token of our OAuth2 specification to see the automatically-included POSTed key-value pairs (such as patient and state), that will be extended by the inclusion of any launch "tokens" below.

Before using a launch context token, consider whether it will be more appropriate to pull this information from an API call instead. Many of the values below can be pulled from common FHIR resources like Patient, Practitioner, Encounter, and more.

Contexts & Launch Points

Many tokens require users to be in a specific workflow or **Context** to function. For example, you can't launch with a patient's MRN unless the workflow has the context of a specific patient, such as the patient's chart or a specific lab result. To help you find tokens relevant for your workflows, this document has separate sections/tables for each context.

Some tokens only work in specific Epic applications, such as our primary desktop app Hyperspace, our provider-facing mobile apps like Haiku, Canto, or Rover, or our patient-facing application, MyChart. If a token is restricted to specific apps, they will be noted in the **Launch Points** column.

Any Context

The following tokens are not restricted to any specific workflow context

Token	Context	Launch Points	Description
%CLIENTHOSTSOURCE%	Any	Hyperspace	The host name of the frame where the application is hosted. This can be used to add X-Frame-Options to the allowlist.
%ENVIRONMENTTYPE%	Any		The name of the Epic environment from which the app is launched.
%EPICUSERFHIRID%	Any	Hyperspace	The user's FHIR ID in Epic. This token returns a value for all users, even without a linked provider record.
%EPICUSERID%	Any	Hyperspace	The user's ID in Epic.
%EPICVERSION%	Any		A numeric representation of the Epic version of the software from which the app is launched.
%HUBTOPIC%	Any	Hyperspace	The URL your application's server uses to handle FHIRcast communication workflows. See FHIRcast specs for more details.
%HUBURL%	Any	Hyperspace	The unique topic your application's server uses to handle FHIRcast communication workflows. See FHIRcast specs for more details.
%INSTALLDIR%	Any		The data installation directory. Used primarily in file-based PACS integrations.
%IPADDR%	Any		The IP address of the workstation.
%IPV6ADDR%	Any		The IPv6 address of the workstation or, if the PACS vendor doesn't support IPv6, the IPv4 address.
%LOGINDEPTID%	Any		The ID of the authenticated user's login department



Token Library

EHR Launch

Token Library

Health system analysts can configure an integration record to launch a URL, and when doing so, they can use a library of "tokens," which are substituted with identifiers from the user's context upon the launch of your application. The context column notes the minimum amount of information that must be available to the app at the time of launch to ensure data is populated.

In an OAuth2 SMART on FHIR workflow, these "tokens" are sent as separate key-value pairs, POSTed in the same response with (but not included in) the access token. Refer to Step 5: Your Application Exchanges the Authorization Code for an Access Token of our OAuth2 specification to see the automatically-included POSTed key-value pairs (such as patient and state), that will be extended by the inclusion of any launch "tokens" below.

Before using a launch context token, consider whether it will be more appropriate to pull this information from an API call instead. Many of the values below can be pulled from common FHIR resources like Patient, Practitioner, Encounter, and more.

Contexts & Launch Points

Many tokens require users to be in a specific workflow or **Context** to function. For example, you can't launch with a patient's MRN unless the workflow has the context of a specific patient, such as the patient's chart or a specific lab result. To help you find tokens relevant for your workflows, this document has separate sections/tables for each context.

Some tokens only work in specific Epic applications, such as our primary desktop app Hyperspace, our provider-facing mobile apps like Haiku, Canto, or Rover, or our patient-facing application, MyChart. If a token is restricted to specific apps, they will be noted in the **Launch Points** column.

Any Context

The following tokens are not restricted to any specific workflow context

Token	Context	Launch Points	Description
%CLIENTHOSTSOURCE%	Any	Hyperspace	The host name of the frame where the application is hosted. This can be used to add X-Frame-Options to the allowlist.
%ENVIRONMENTTYPE%	Any		The name of the Epic environment from which the app is launched.
%EPICUSERFHIRID%	Any	Hyperspace	The user's FHIR ID in Epic. This token returns a value for all users, even without a linked provider record.
%EPICUSERID%	Any	Hyperspace	The user's ID in Epic.
%EPICVERSION%	Any		A numeric representation of the Epic version of the software from which the app is launched.
%HUBTOPIC%	Any	Hyperspace	The URL your application's server uses to handle FHIRcast communication workflows. See FHIRcast specs for more details.
%HUBURL%	Any	Hyperspace	The unique topic your application's server uses to handle FHIRcast communication workflows. See FHIRcast specs for more details.
%INSTALLDIR%	Any		The data installation directory. Used primarily in file-based PACS integrations.
%IPADDR%	Any		The IP address of the workstation.
%IPV6ADDR%	Any		The IPv6 address of the workstation or, if the PACS vendor doesn't support IPv6, the IPv4 address.
%LOGINDEPTID%	Any		The ID of the authenticated user's login department



Token Library

EHR Launch

Token Library

Health system analysts can configure an integration record to launch a URL, and when doing so, they can use a library of "tokens," which are substituted with identifiers from the user's context upon the launch of your application. The context column notes the minimum amount of information that must be available to the app at the time of launch to ensure data is populated.

In an OAuth2 SMART on FHIR workflow, these "tokens" are sent as separate key-value pairs, POSTed in the same response with (but not included in) the access token. Refer to Step 5: Your Application Exchanges the Authorization Code for an Access Token of our OAuth2 specification to see the automatically-included POSTed key-value pairs (such as patient and state), that will be extended by the inclusion of any launch "tokens" below.

Before using a launch context token, consider whether it will be more appropriate to pull this information from an API call instead. Many of the values below can be pulled from common FHIR resources like Patient, Practitioner, Encounter, and more.

Contexts & Launch Points

Many tokens require users to be in a specific workflow or **Context** to function. For example, you can't launch with a patient's MRN unless the workflow has the context of a specific patient, such as the patient's chart or a specific lab result. To help you find tokens relevant for your workflows, this document has separate sections/tables for each context.

Some tokens only work in specific Epic applications, such as our primary desktop app Hyperspace, our provider-facing mobile apps like Haiku, Canto, or Rover, or our patient-facing application, MyChart. If a token is restricted to specific apps, they will be noted in the **Launch Points** column.

Any Context

The following tokens are not restricted to any specific workflow context

Token	Context	Launch Points	Description
%CLIENTHOSTSOURCE%	Any	Hyperspace	The host name of the frame where the application is hosted. This can be used to add X-Frame-Options to the allowlist.
%ENVIRONMENTTYPE%	Any		The name of the Epic environment from which the app is launched.
%EPICUSERFHIRID%	Any	Hyperspace	The user's FHIR ID in Epic. This token returns a value for all users, even without a linked provider record.
%EPICUSERID%	Any	Hyperspace	The user's ID in Epic.
%EPICVERSION%	Any		A numeric representation of the Epic version of the software from which the app is launched.
%HUBTOPIC%	Any	Hyperspace	The URL your application's server uses to handle FHIRcast communication workflows. See FHIRcast specs for more details.
%HUBURL%	Any	Hyperspace	The unique topic your application's server uses to handle FHIRcast communication workflows. See FHIRcast specs for more details.
%INSTALLDIR%	Any		The data installation directory. Used primarily in file-based PACS integrations.
%IPADDR%	Any		The IP address of the workstation.
%IPV6ADDR%	Any		The IPv6 address of the workstation or, if the PACS vendor doesn't support IPv6, the IPv4 address.
%LOGINDEPTID%	Any		The ID of the authenticated user's login department



Token Library

Token Request OIDC

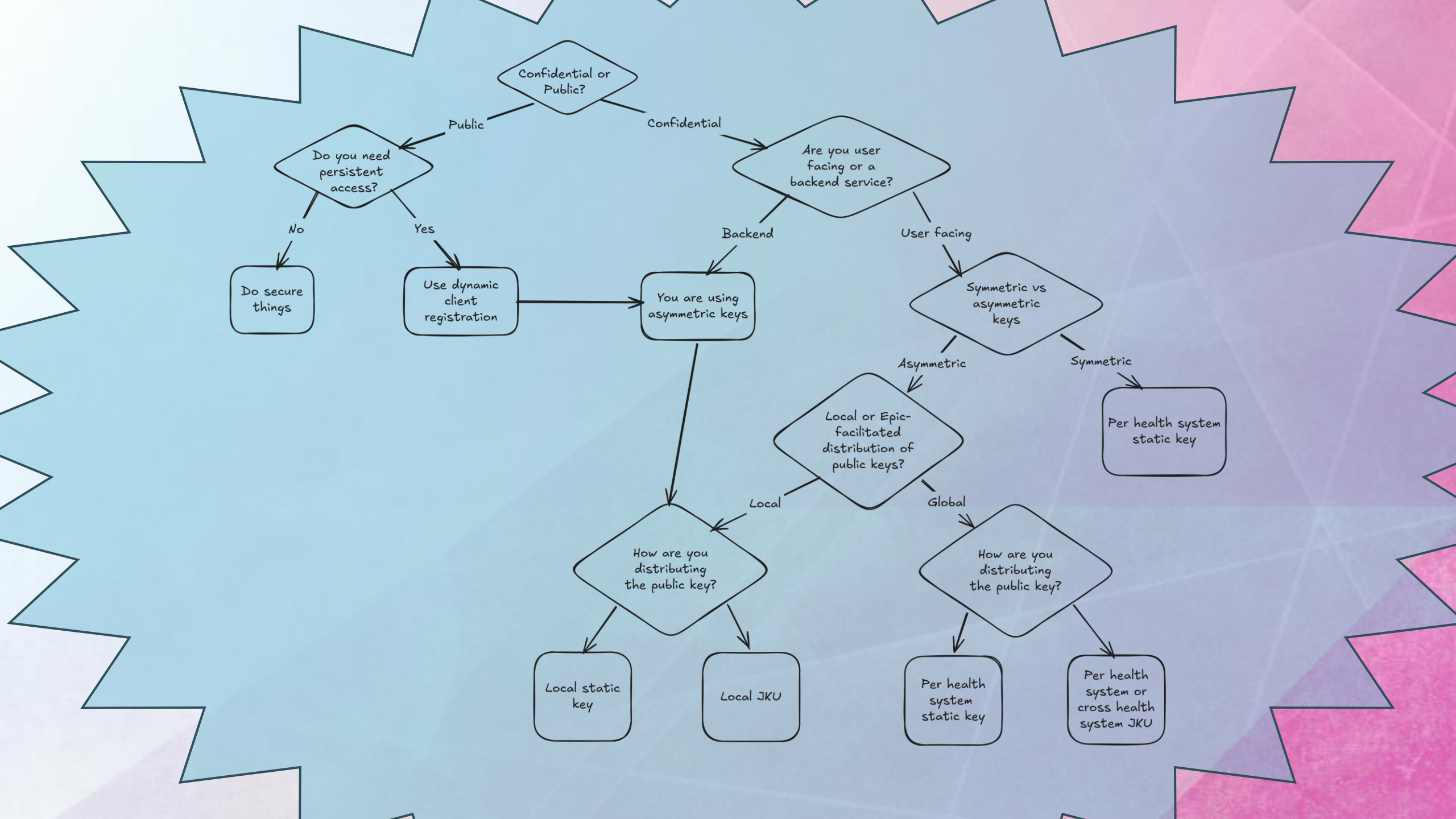

```
{
    "aud": "de5dae1a-4317-4c25-86f1-ed558e85529b",
    "exp": 1595956317,
    "iat": 1595956017,
    "iss": "https://fhir.epic.com/interconnect-fhir-
oauth/oauth2",
    "sub": "exfo6E4EXjWsnhA1OGVElgw3“,
    "nonce": "5xXdlZSeqrAB5aH6cOyPy5zs"
}
```

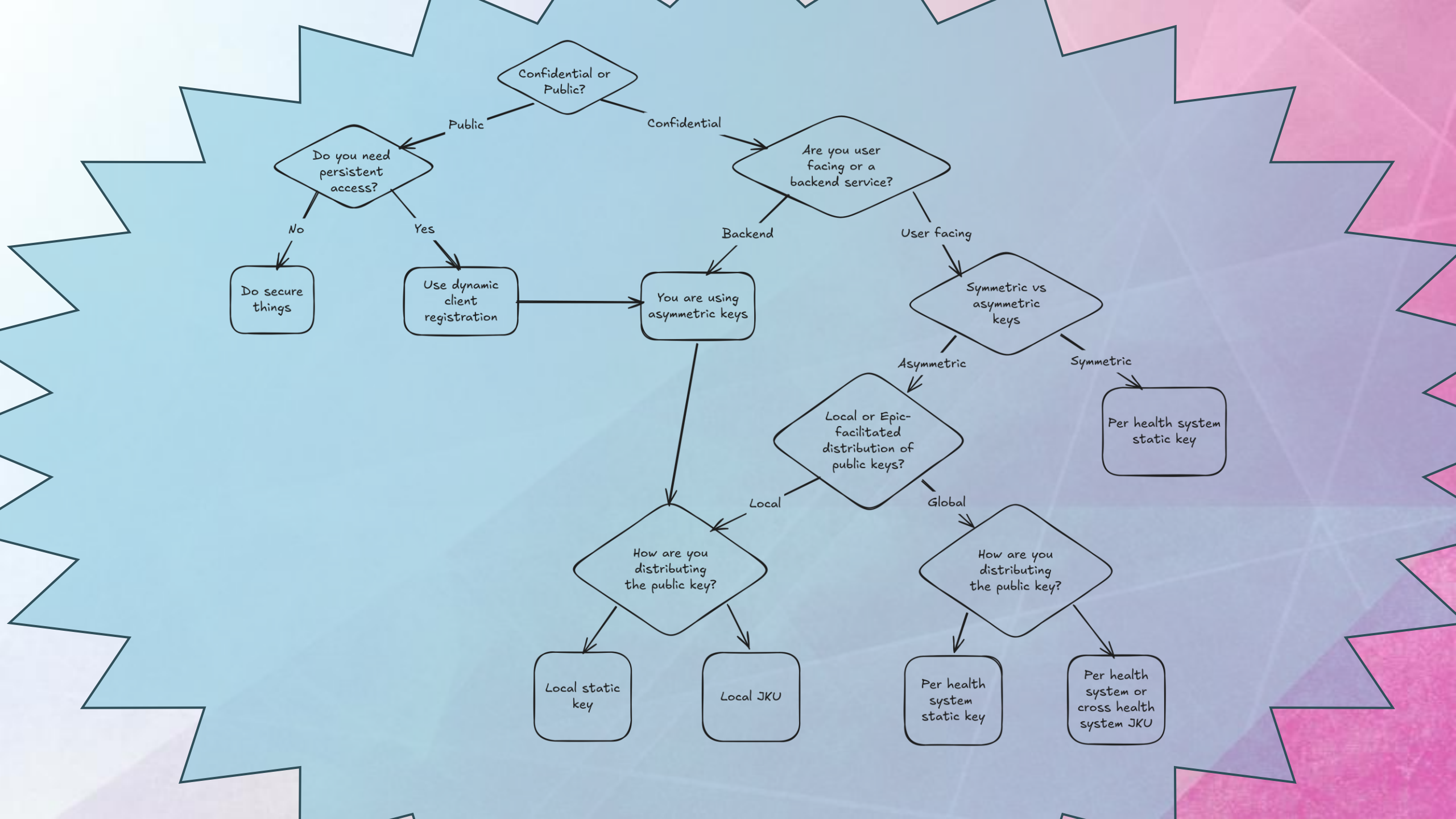



Validating the OpenID Connect JSON Web Token

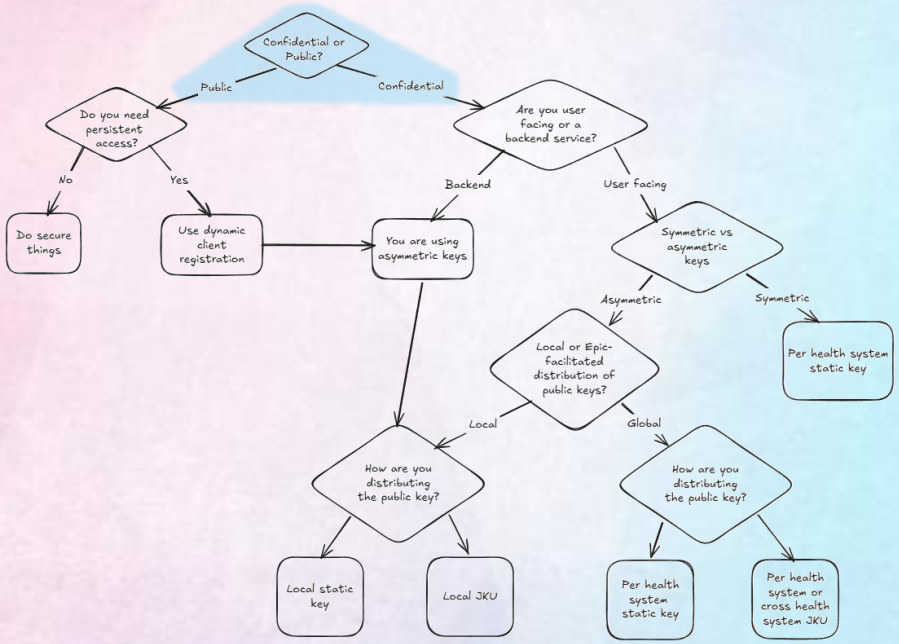
Security Deep Dive

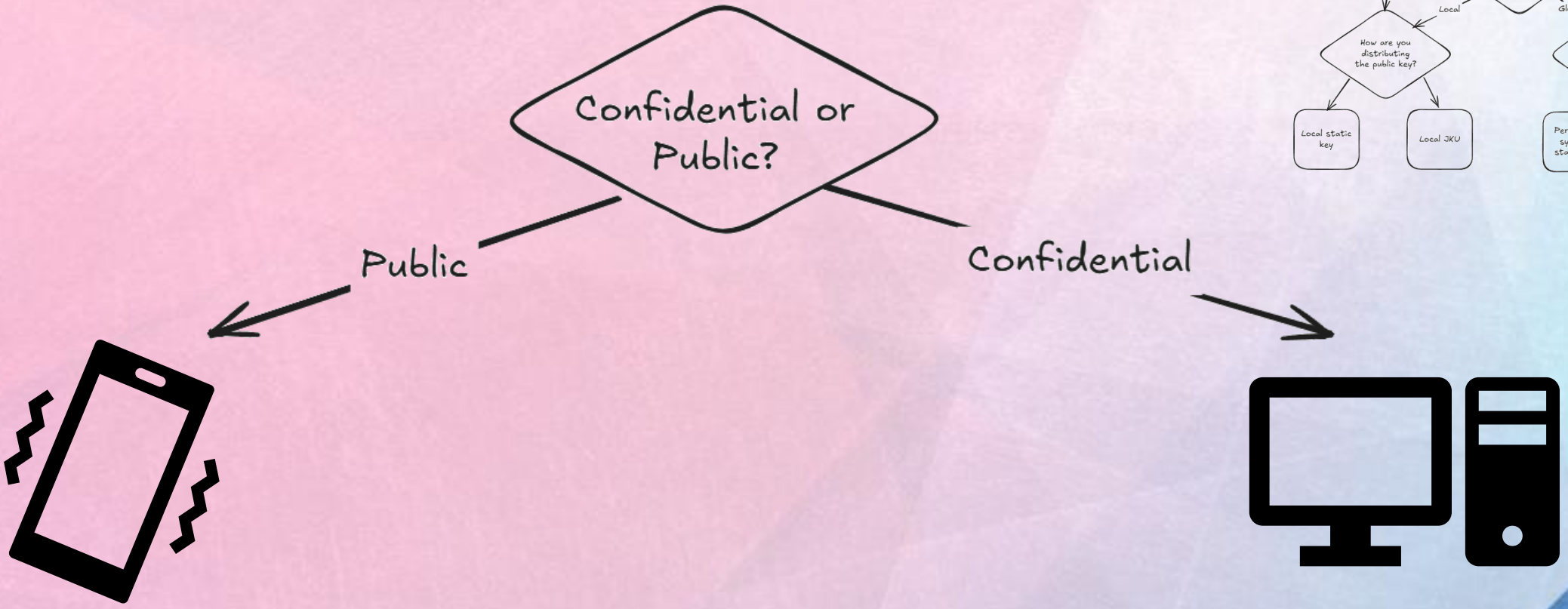



Confidential or Public?
Public
Confidential
Do you need persistent access?
No
Yes
Do secure things
Use dynamic client registration
Are you user facing or a backend service?
Backend
User facing
You are using asymmetric keys
Symmetric vs asymmetric keys
Asymmetric
Symmetric
Per health system static key
Local or Epic-facilitated distribution of public keys?
Local
Global
How are you distributing the public key?
Local static key
Local JKU
How are you distributing the public key?
Per health system static key
Per health system or cross health system JKU

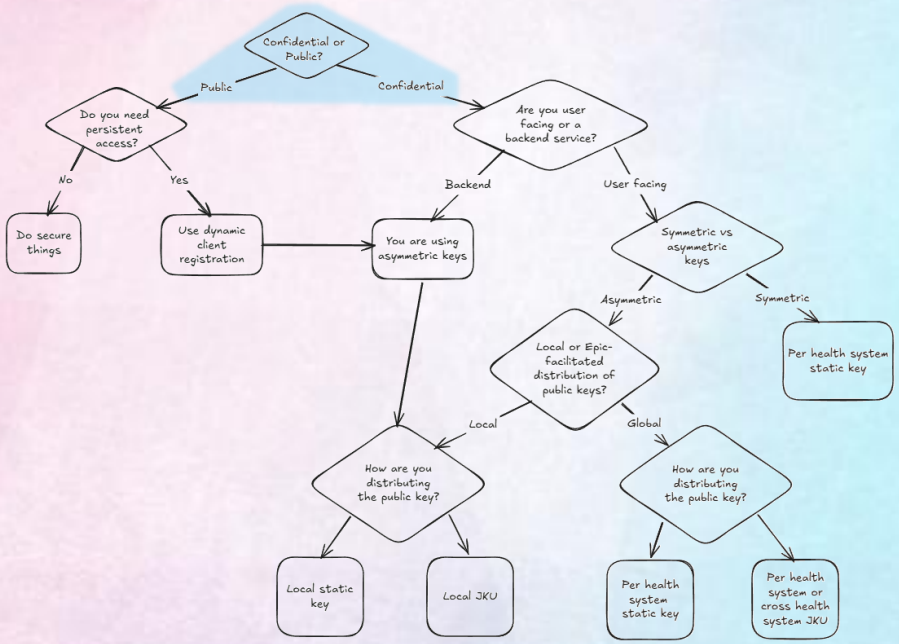

Confidential or Public?
Public
Confidential
Do you need persistent access?
No
Yes
Do secure things
Use dynamic client registration
Are you user facing or a backend service?
Backend
User facing
You are using asymmetric keys
Symmetric vs asymmetric keys
Asymmetric
Symmetric
Per health system static key
Local or Epic-facilitated distribution of public keys?
Local
Global
How are you distributing the public key?
Local static key
Local JKU
How are you distributing the public key?
Per health system static key
Per health system or cross health system JKU

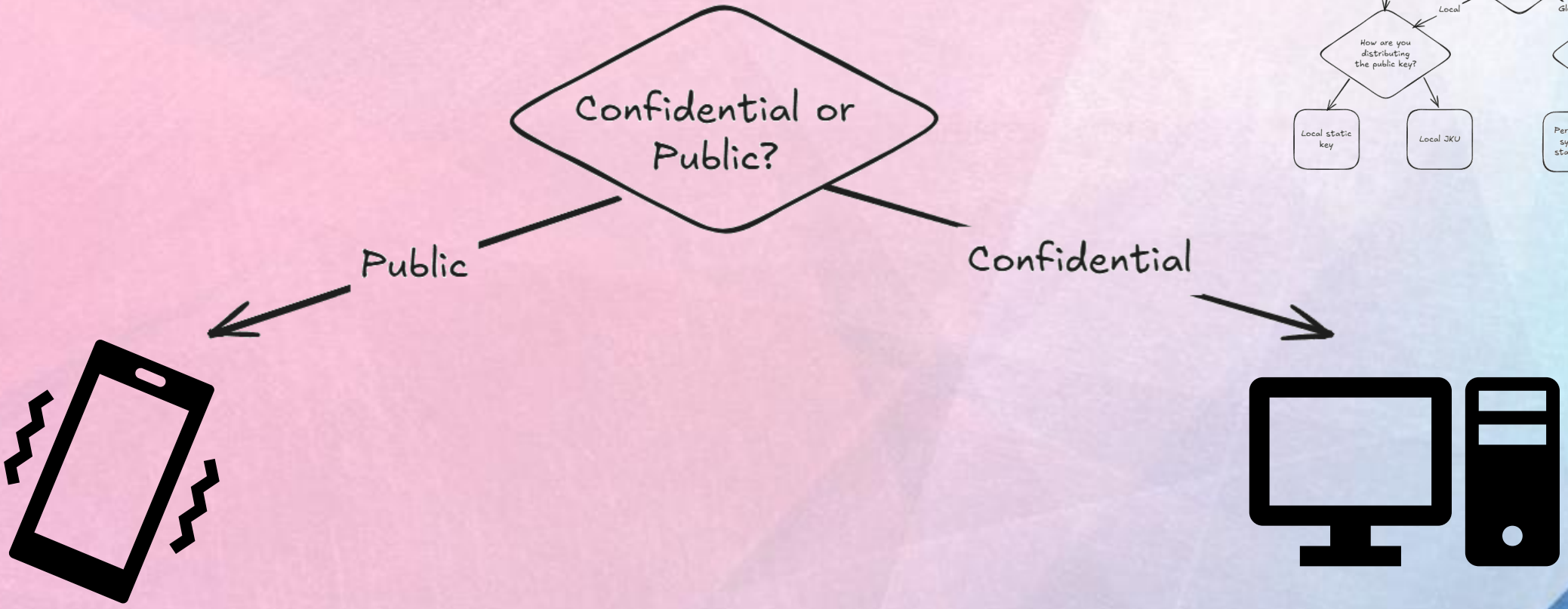
Client Types



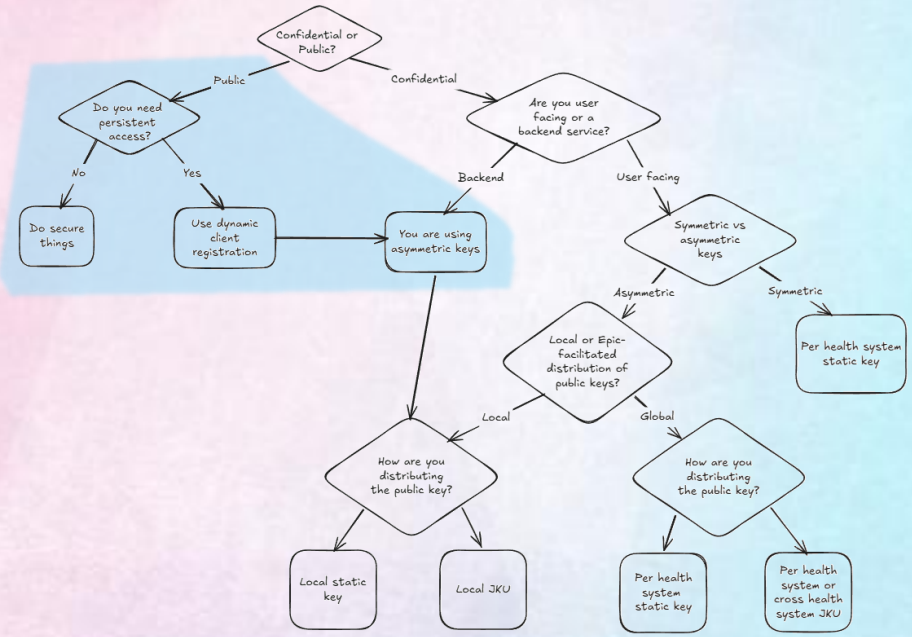


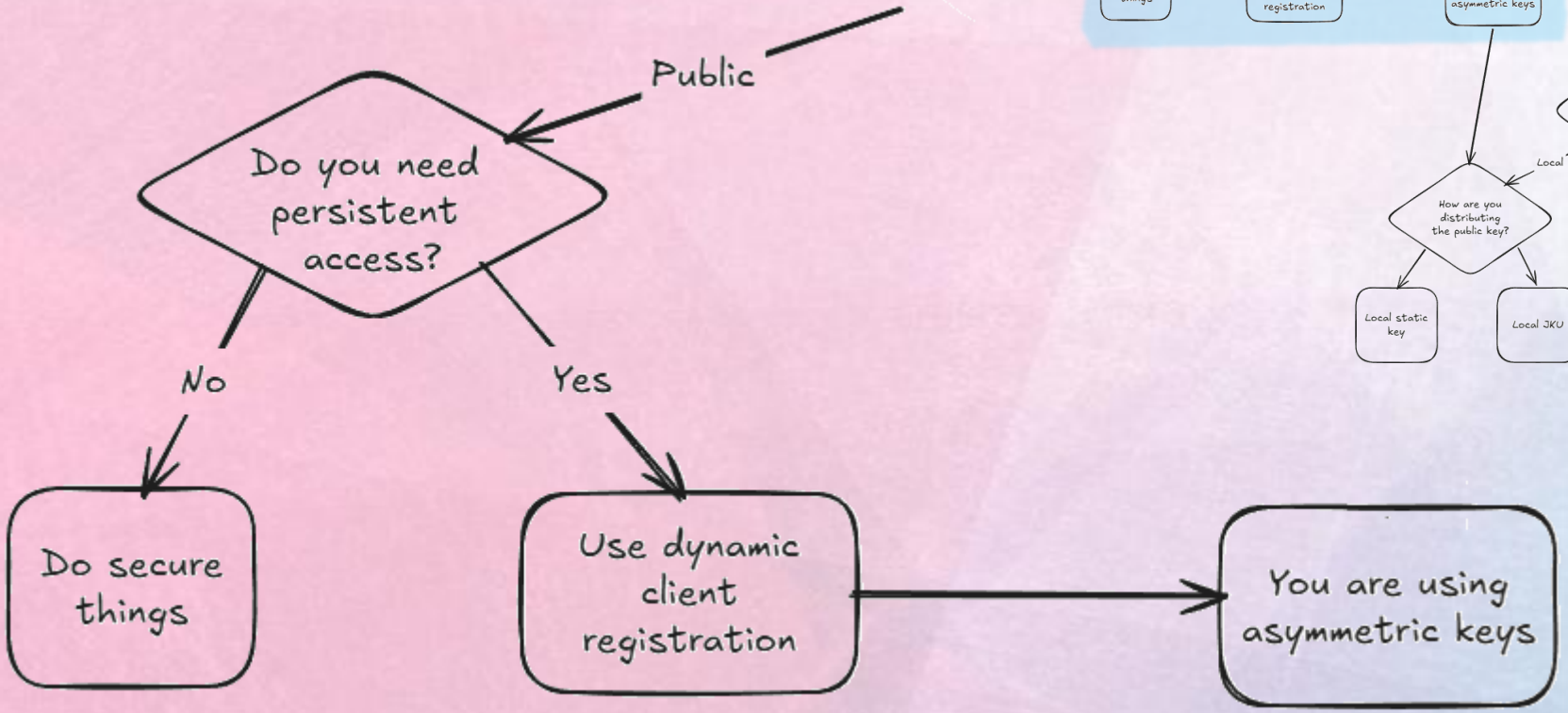
Client Types





Public Client





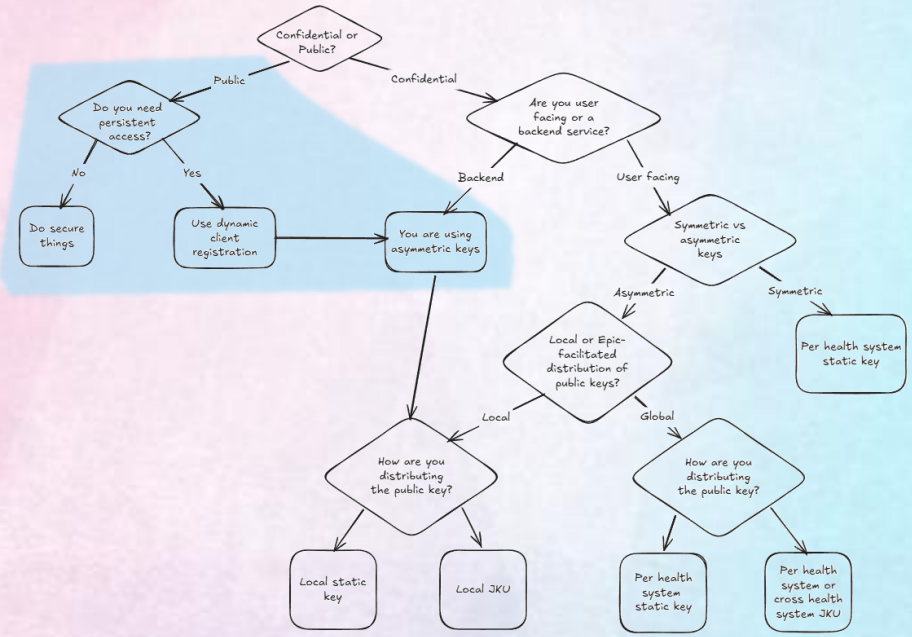


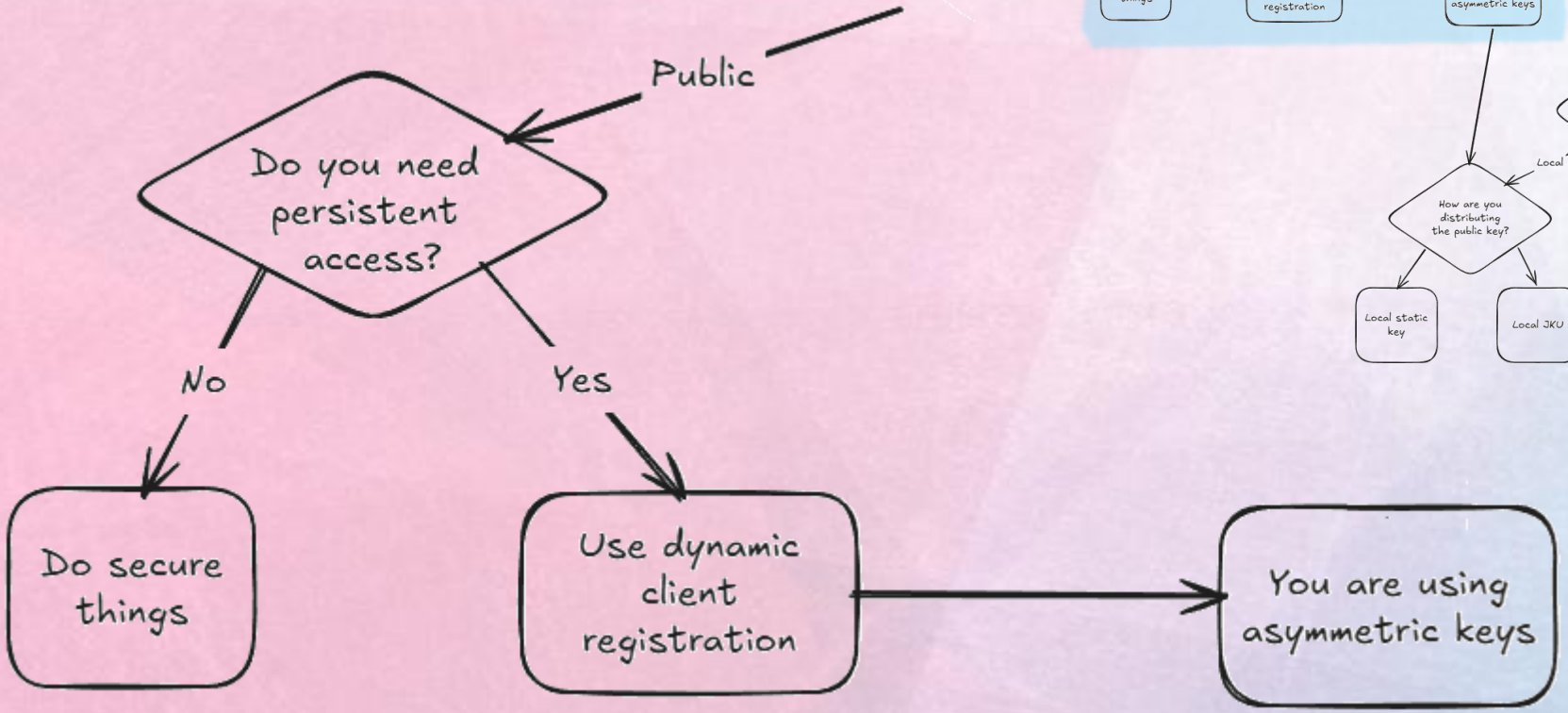
Localhost

PKCE

Dynamic Client

Public Client





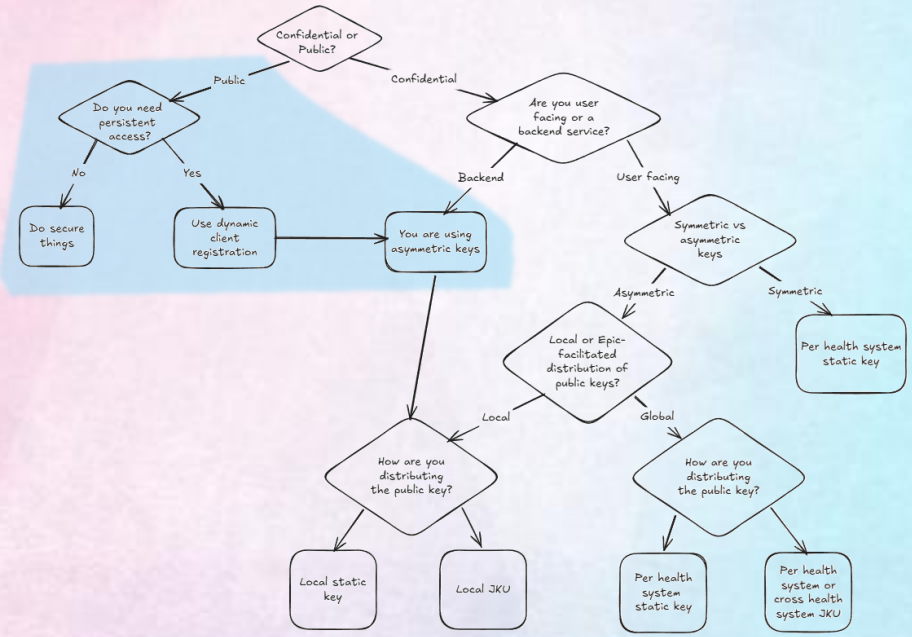


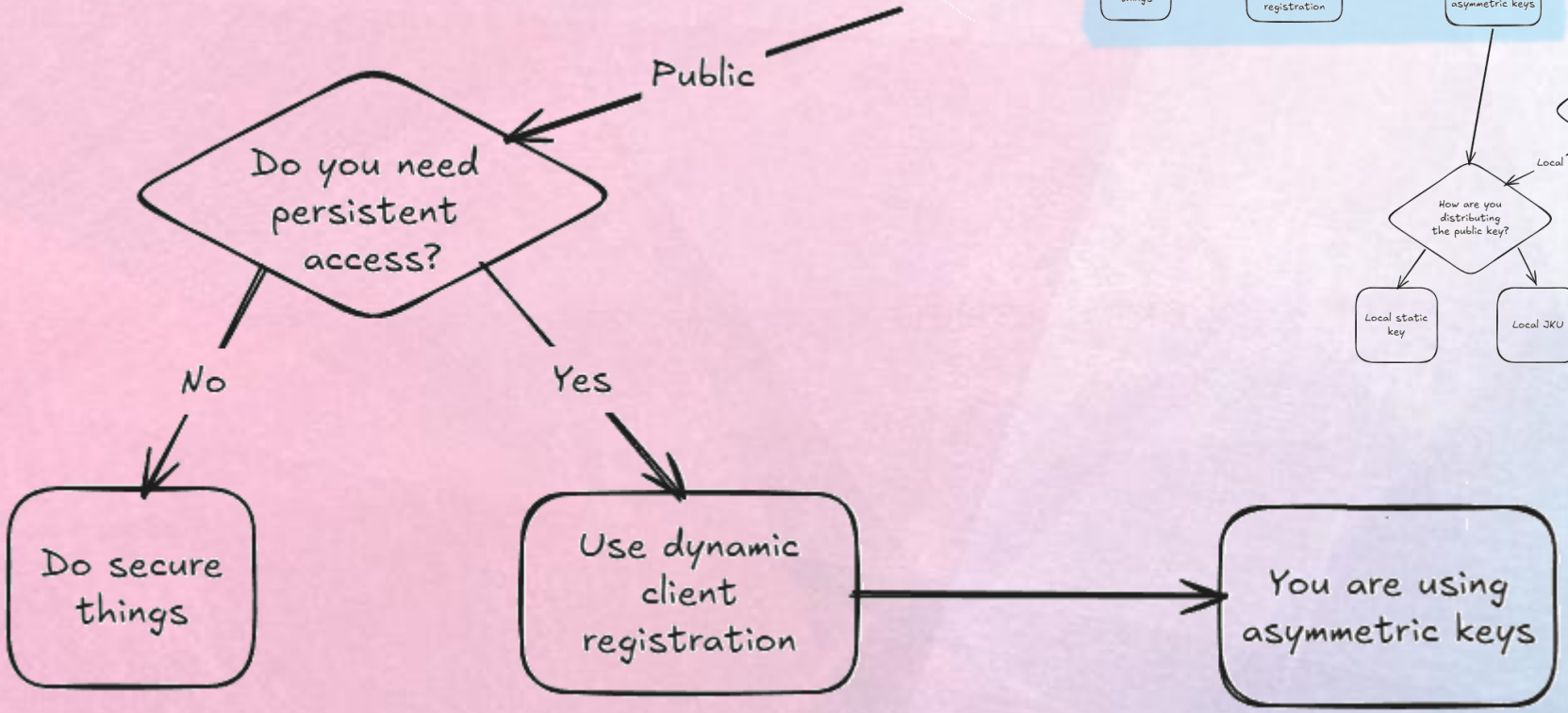
Localhost

PKCE

Dynamic Client

Public Client





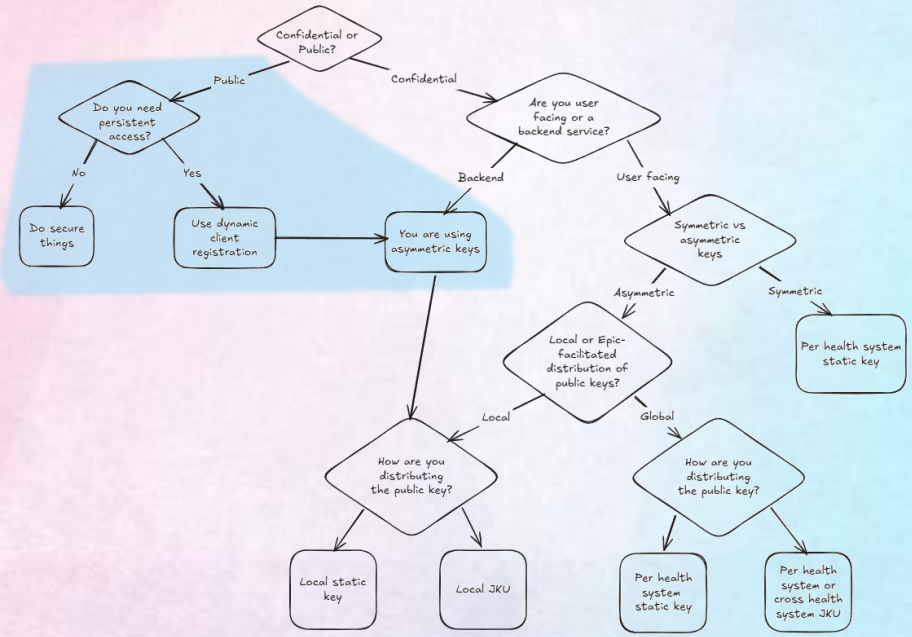


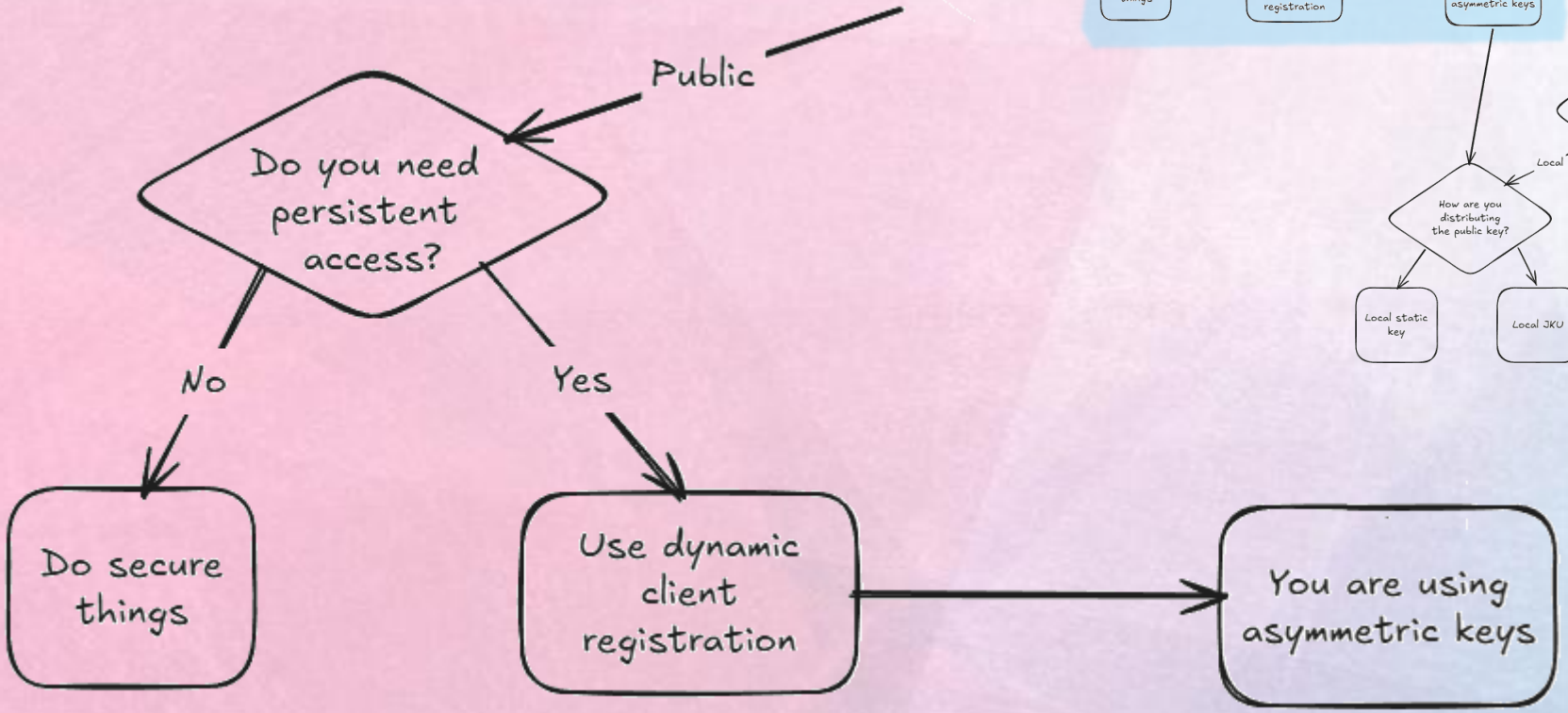
Localhost

PKCE

Dynamic Client

Public Client







Localhost

PKCE

Dynamic Client

User Level Access

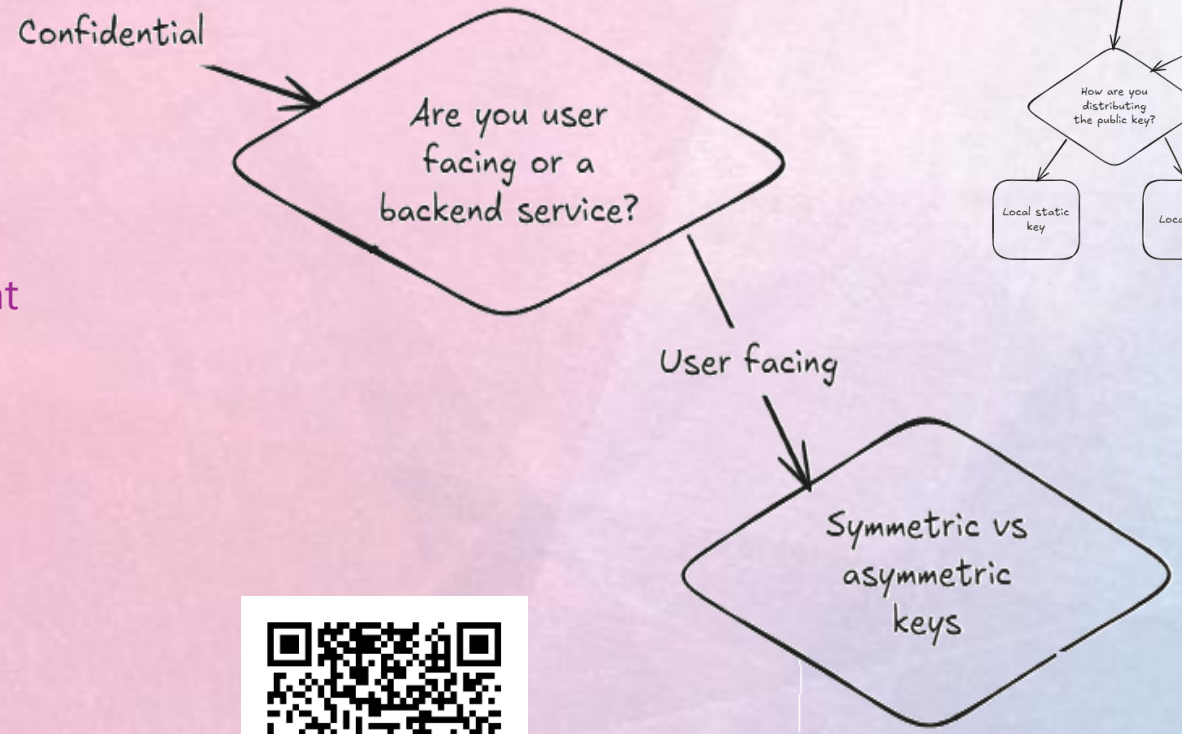
Protected Resource Access

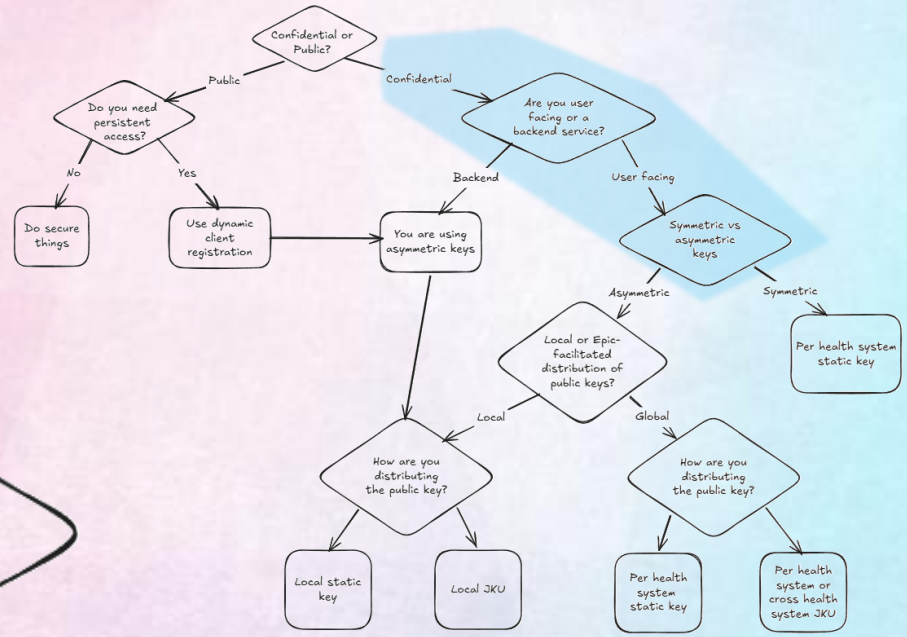


Can view lab results

Can document problems

Can view sensitive notes (e.g. mental health data)





Application User Context

User Level Access

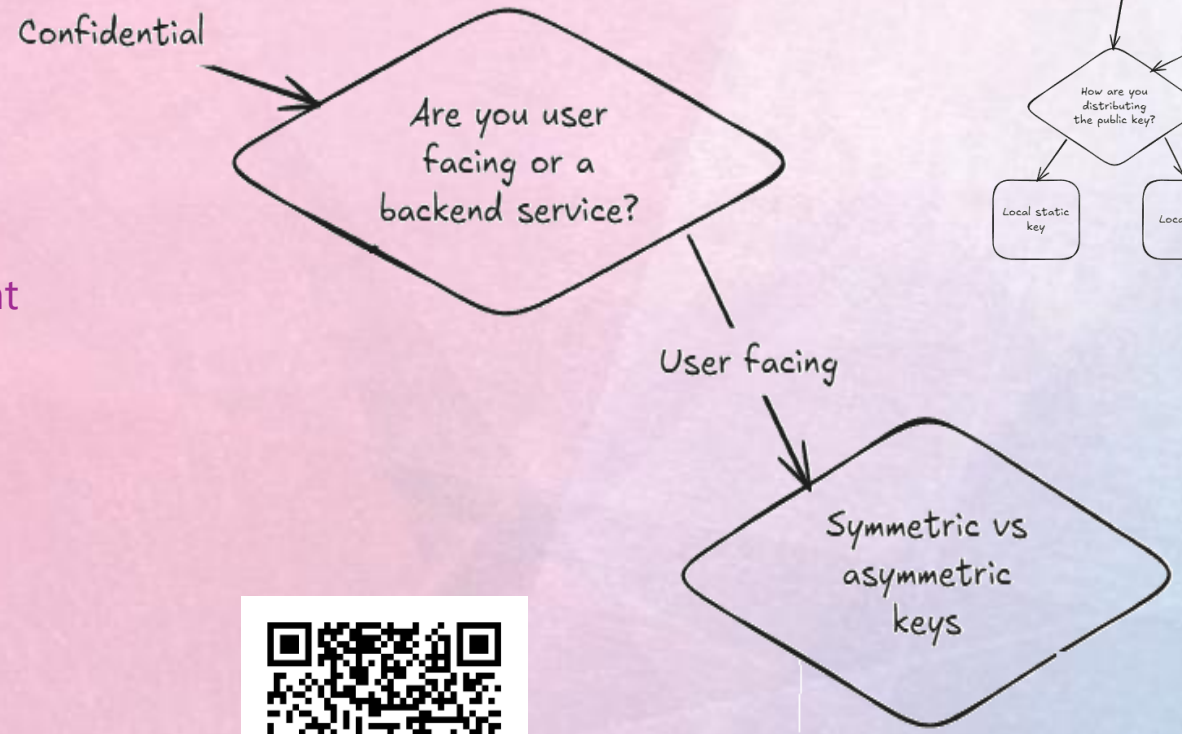
Protected Resource Access

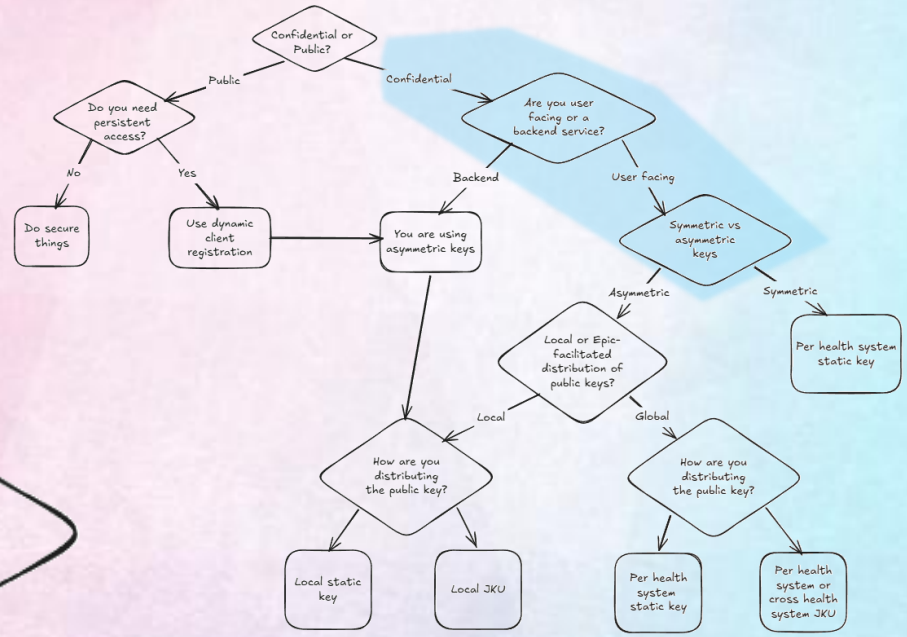


Can view lab results

Can document problems

Can view sensitive notes (e.g. mental health data)





Application User Context

User Level Access

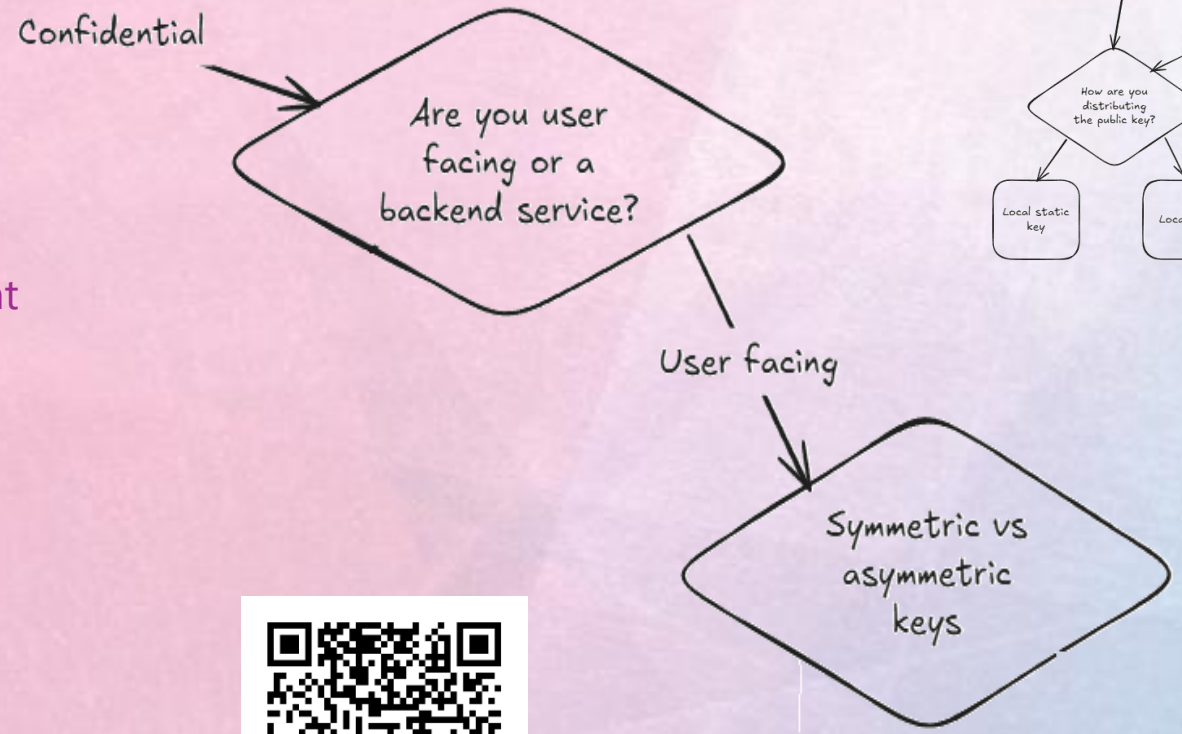
Protected Resource Access

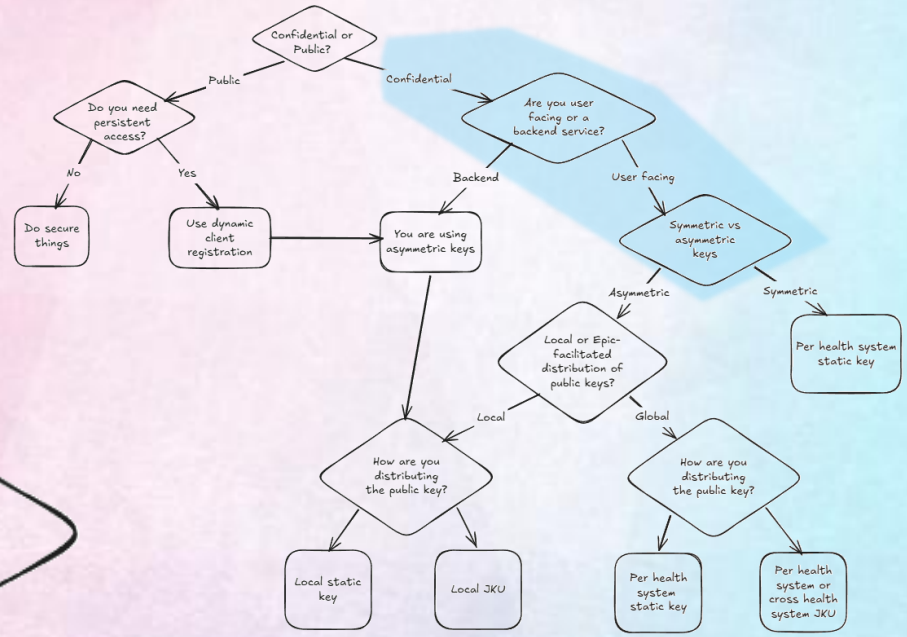


Can view lab results

Can document problems

Can view sensitive notes (e.g. mental health data)





Application User Context

User Level Access

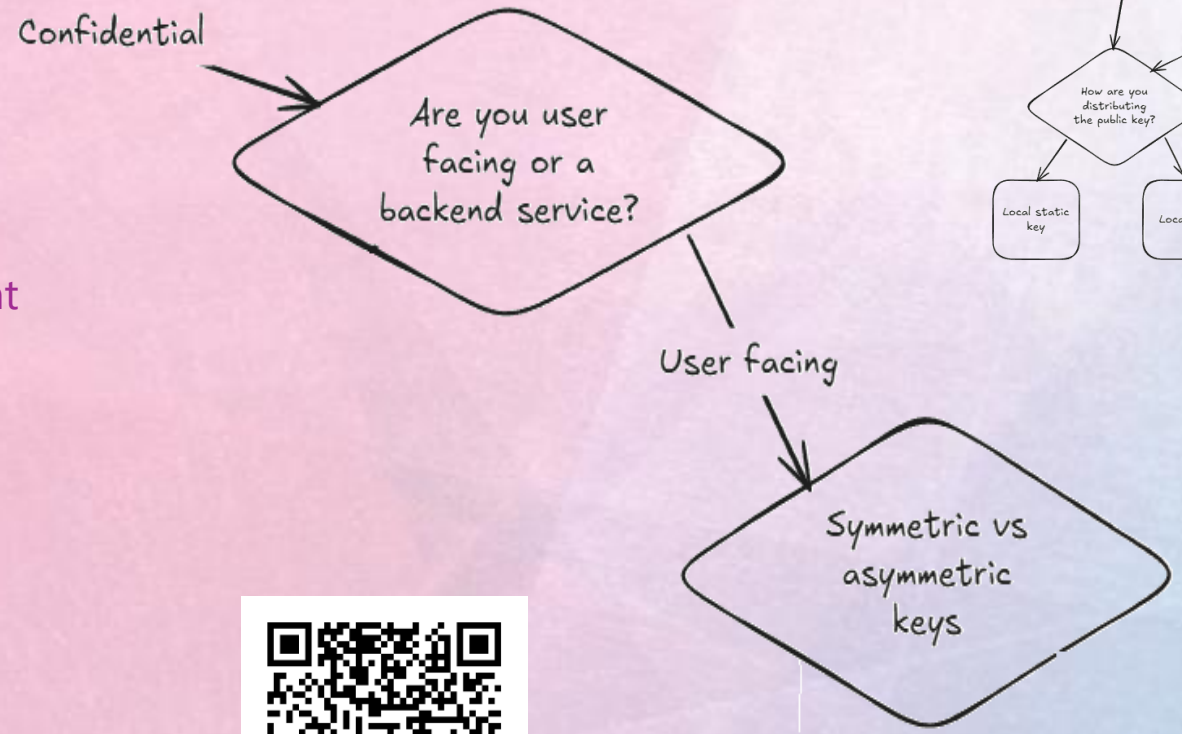
Protected Resource Access

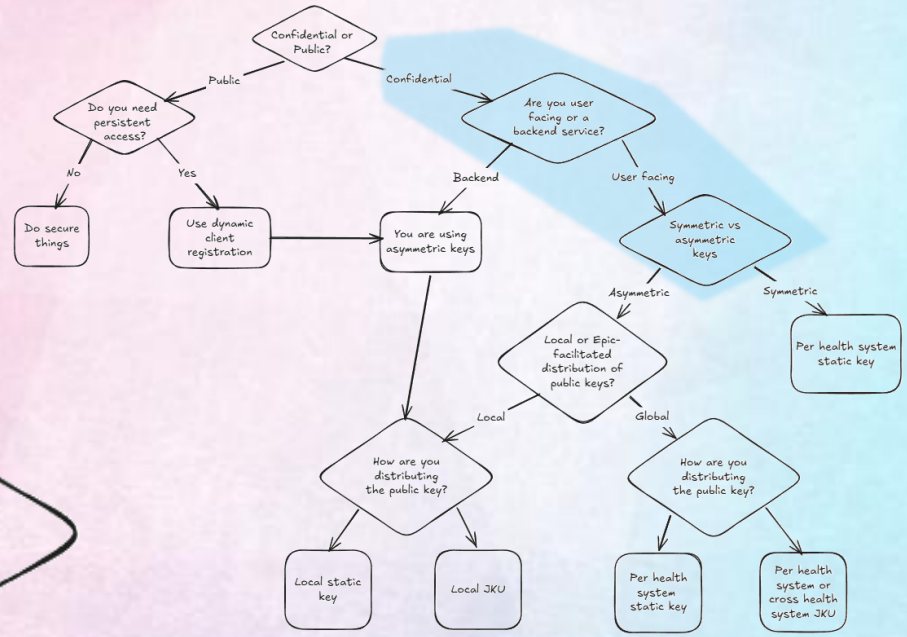


Can view lab results

Can document problems

Can view sensitive notes (e.g. mental health data)





Application User Context

User Level Access

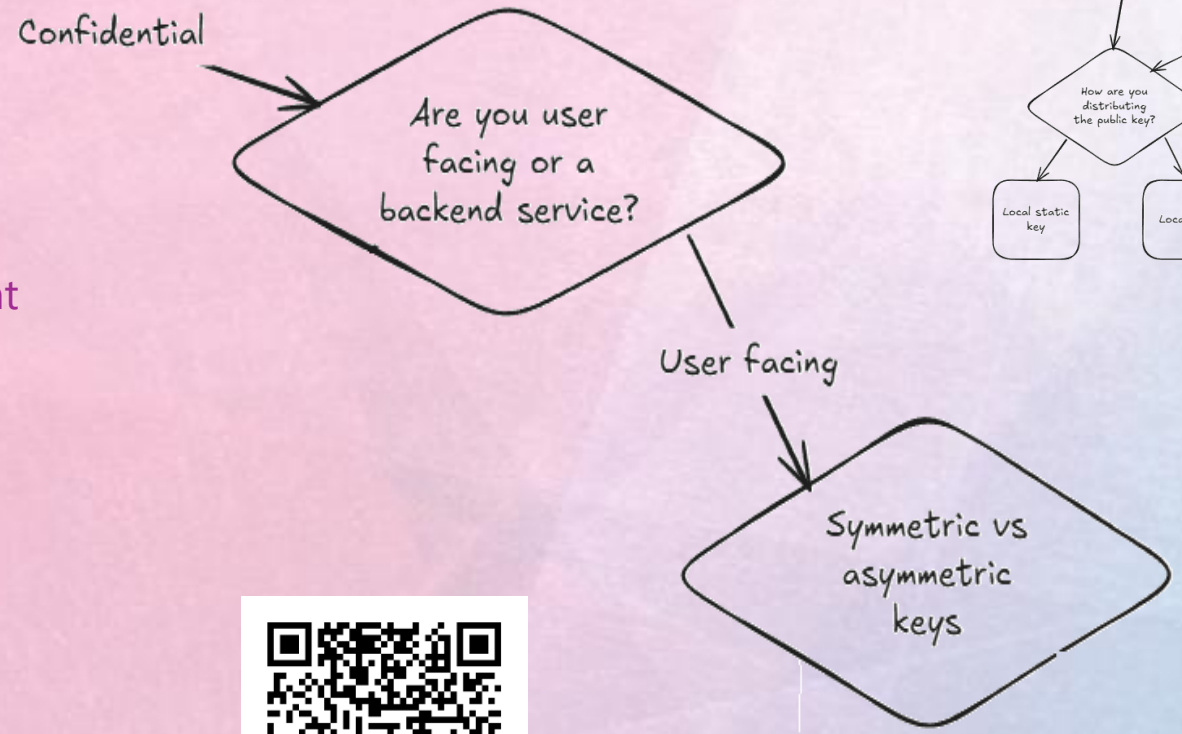
Protected Resource Access

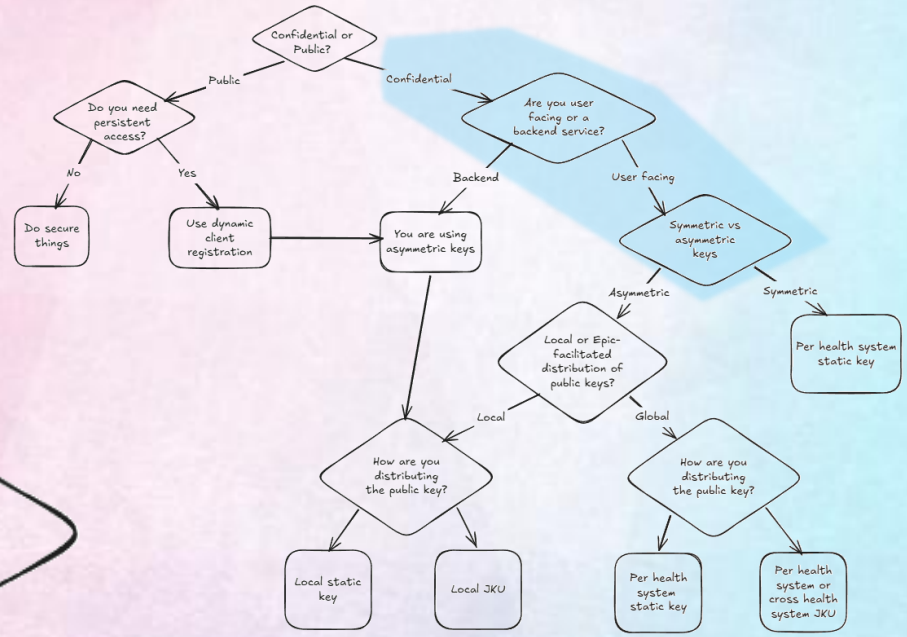


Can view lab results

Can document problems

Can view sensitive notes (e.g. mental health data)

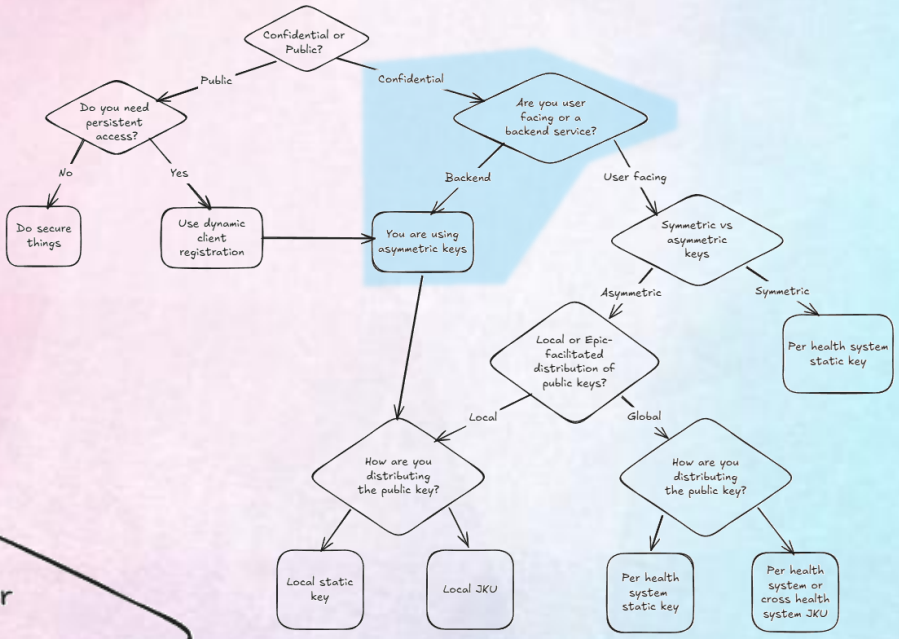




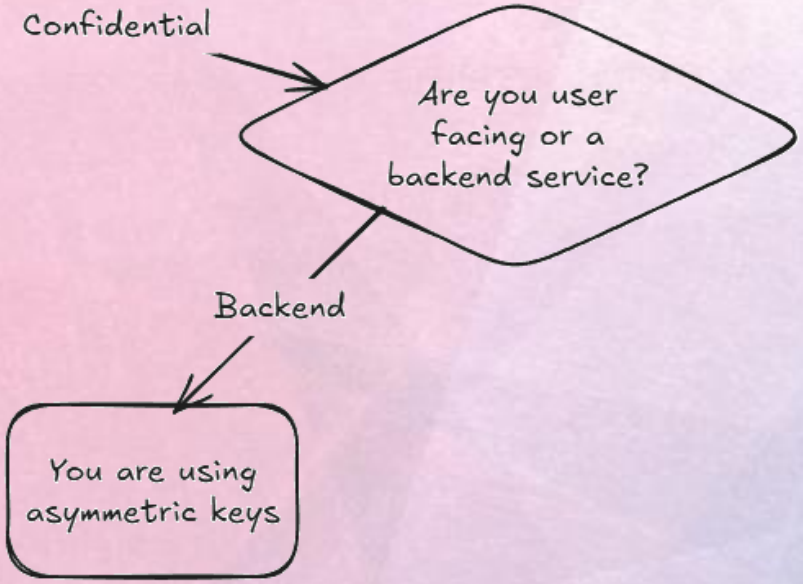
Application User Context

System Level Access

Protected Resource Access









Provisioned User

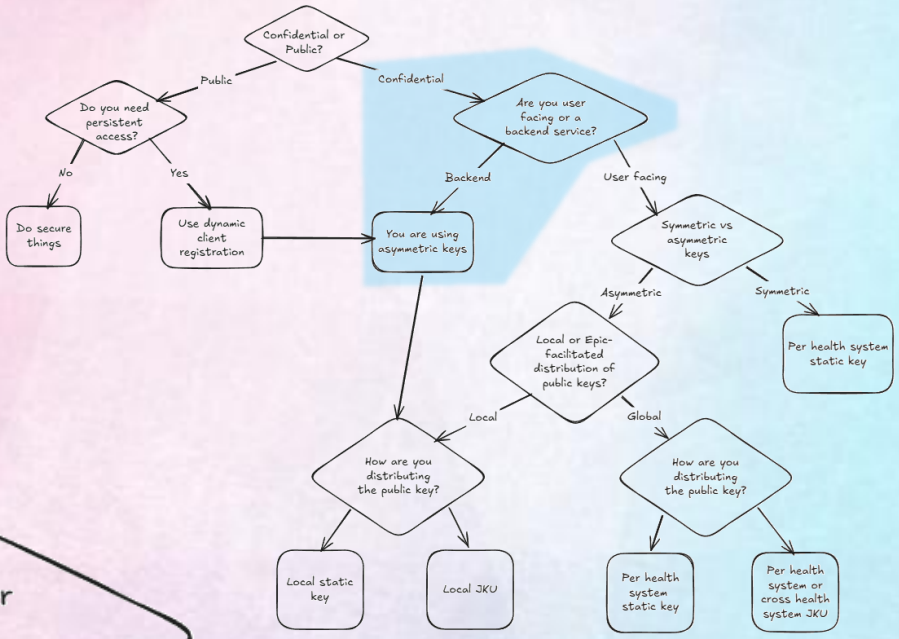
Can view lab results?

Can document problems?

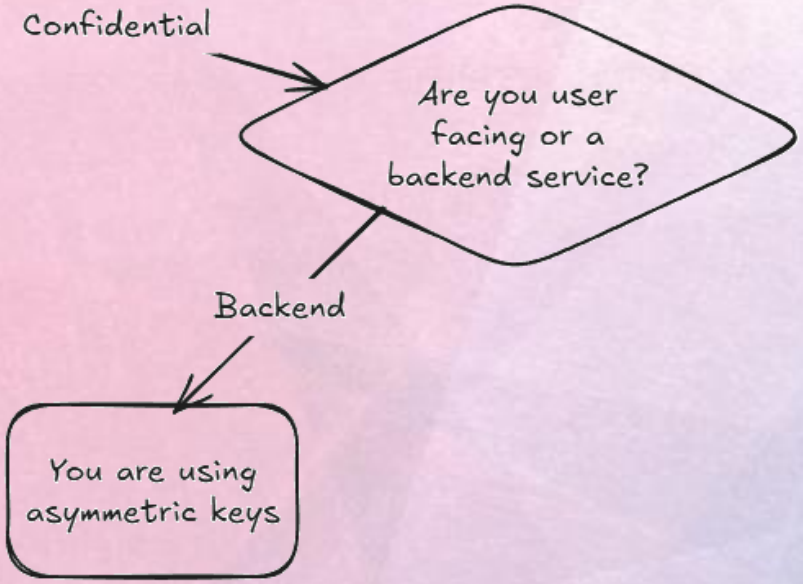
Can view sensitive notes?

System Level Access

Protected Resource Access









Provisioned User

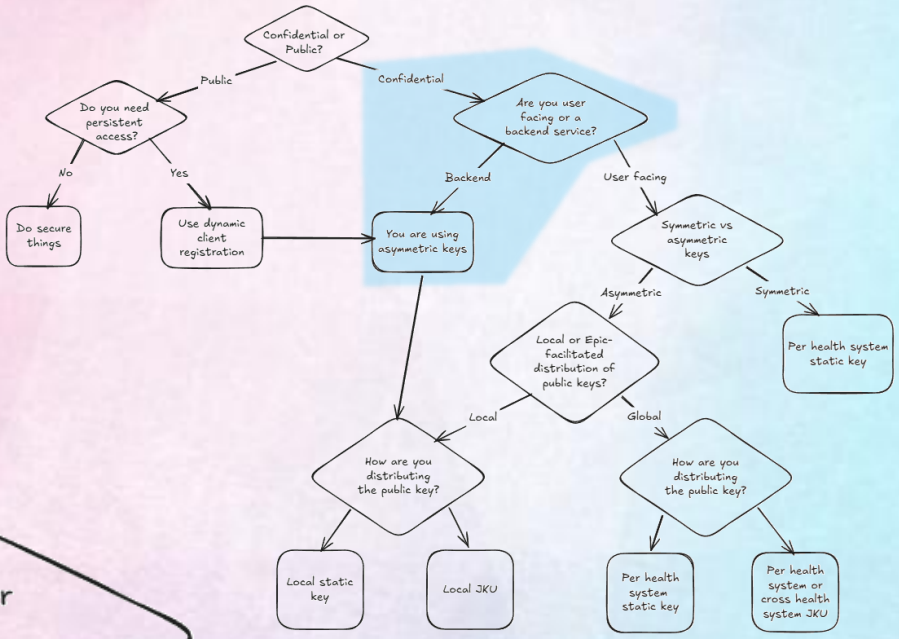
Can view lab results?

Can document problems?

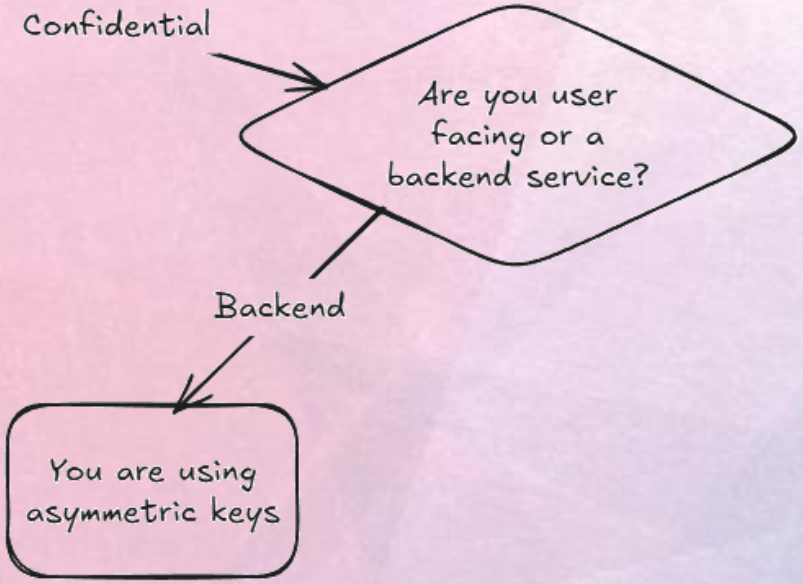
Can view sensitive notes?

System Level Access

Protected Resource Access









Provisioned User

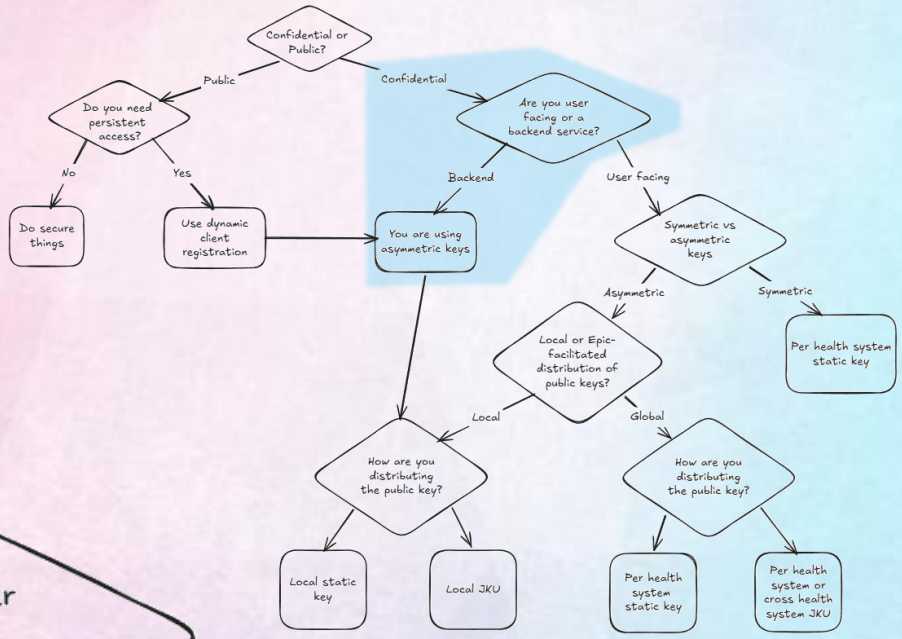
Can view lab results?

Can document problems?

Can view sensitive notes?

System Level Access

Protected Resource Access





Confidential

Are you user facing or a backend service?

Backend

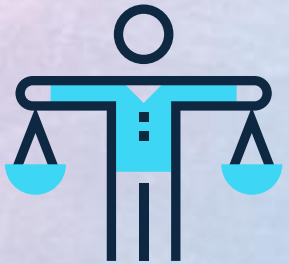
You are using asymmetric keys

Provisioned User

Can view lab results?

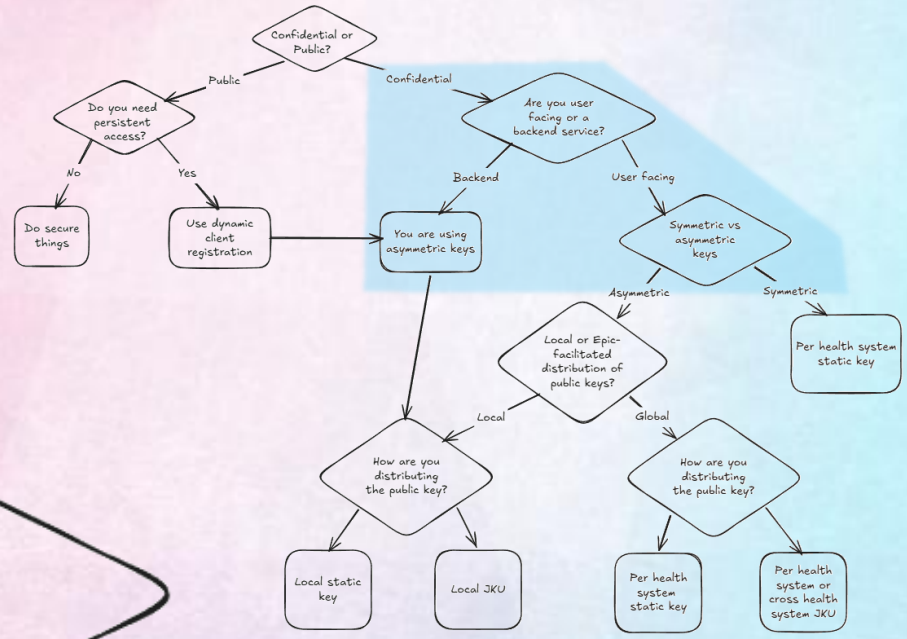
Can document problems?

Can view sensitive notes?





Key Strategy



Symmetric

One Key



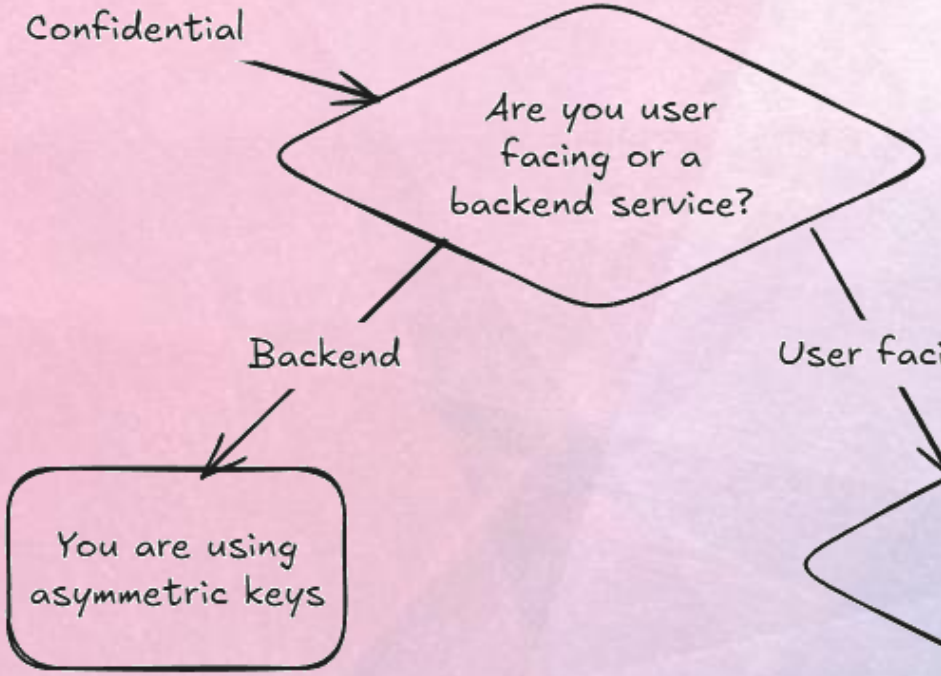
Asymmetric

Public Key

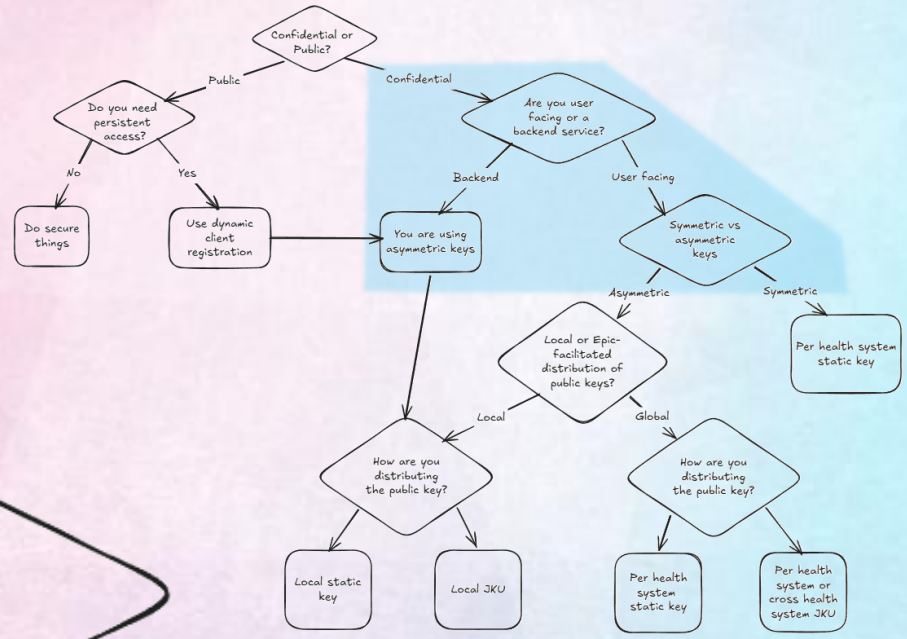


Private Key





Key Strategy



Symmetric

One Key



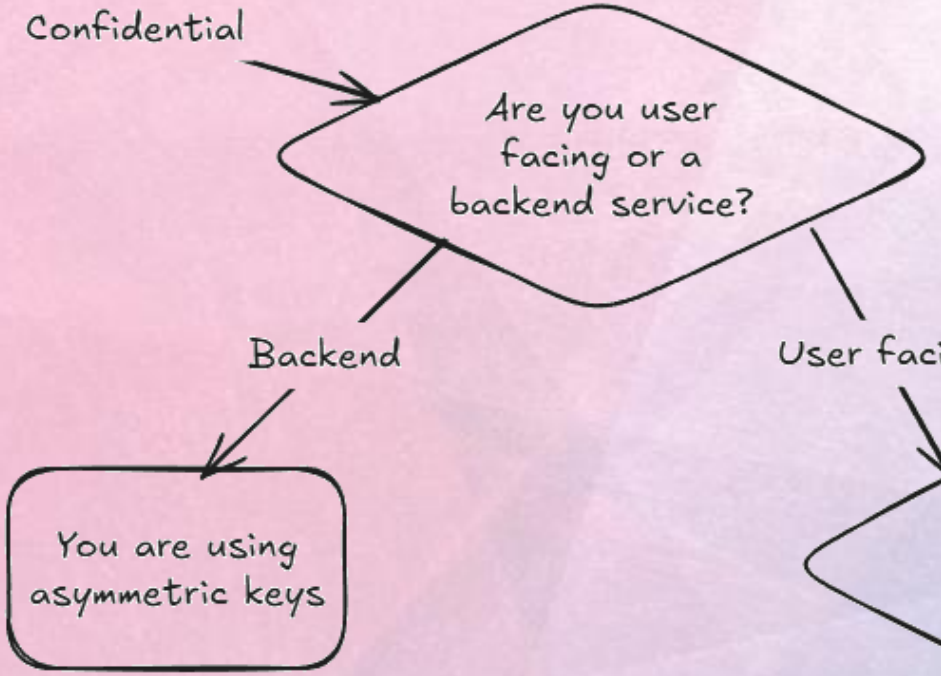
Asymmetric

Public Key

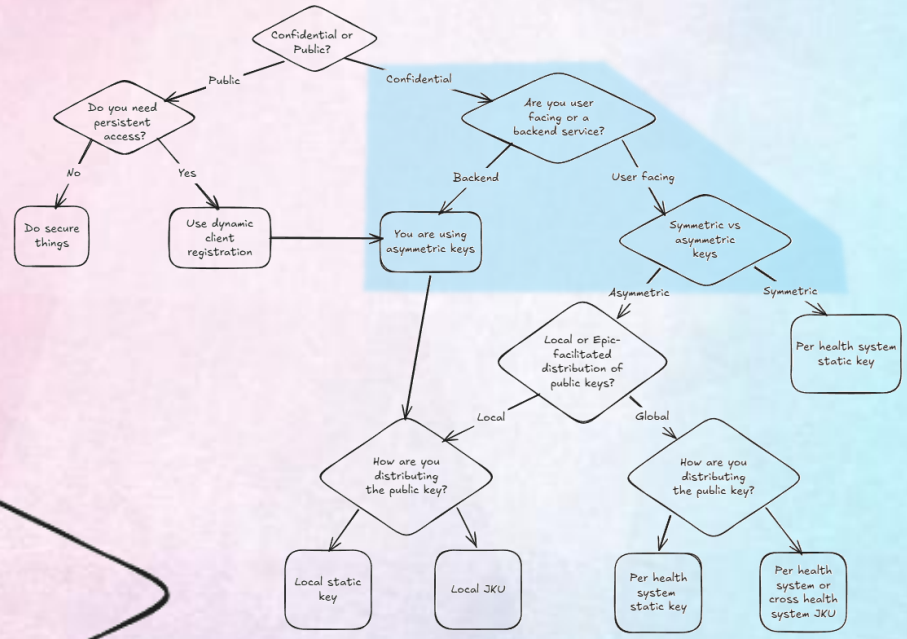


Private Key





Key Strategy



Symmetric

One Key



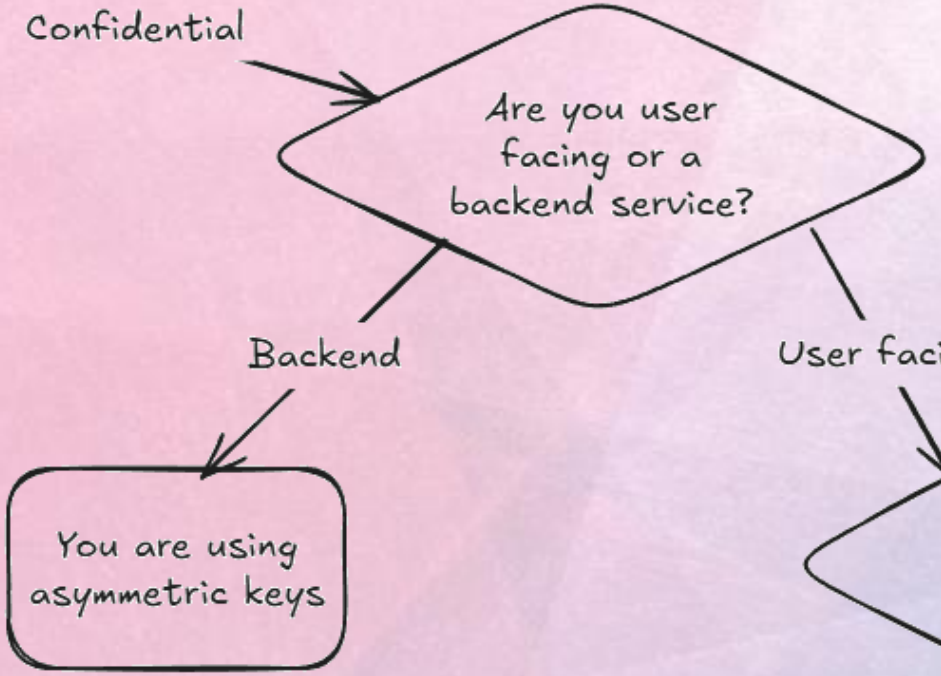
Asymmetric

Public Key

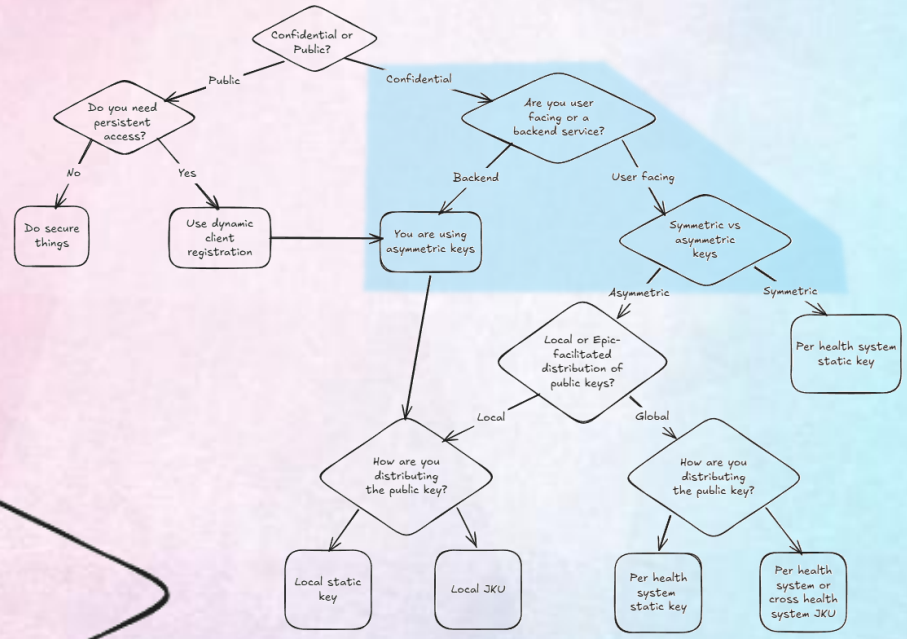


Private Key





Key Strategy



Symmetric

One Key



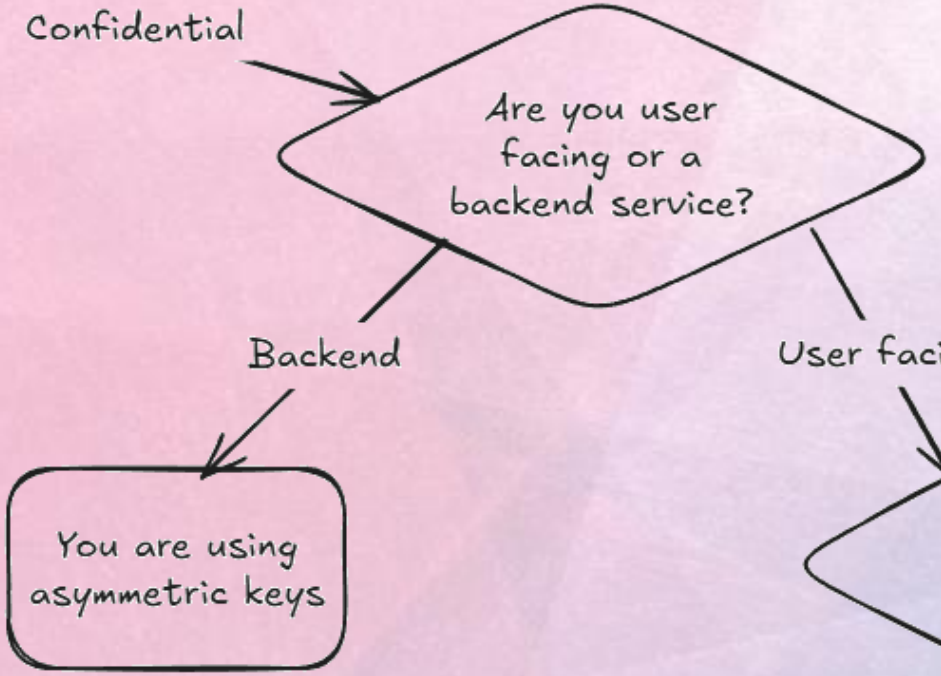
Asymmetric

Public Key



Private Key





Static Symmetric Key



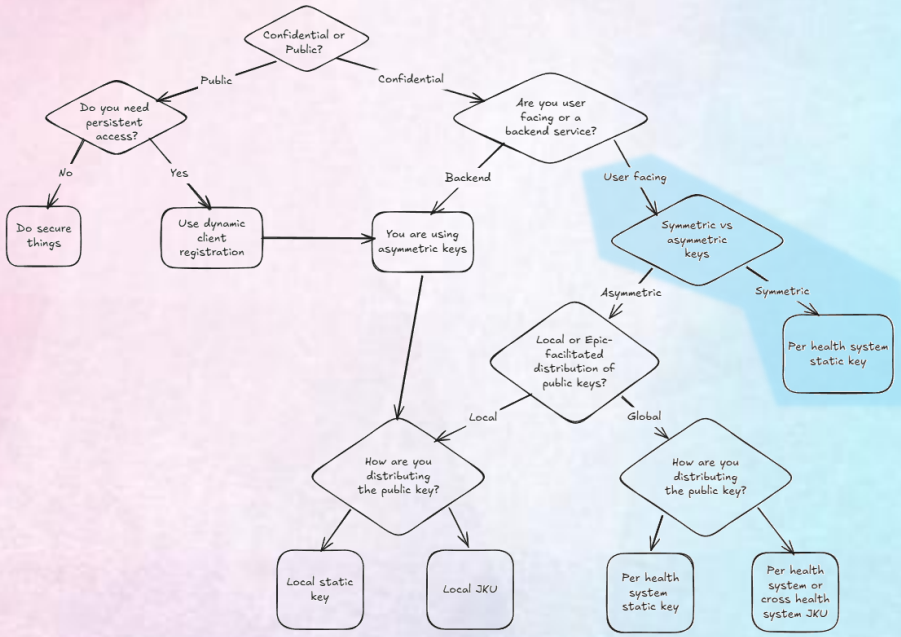
One Key



One Key

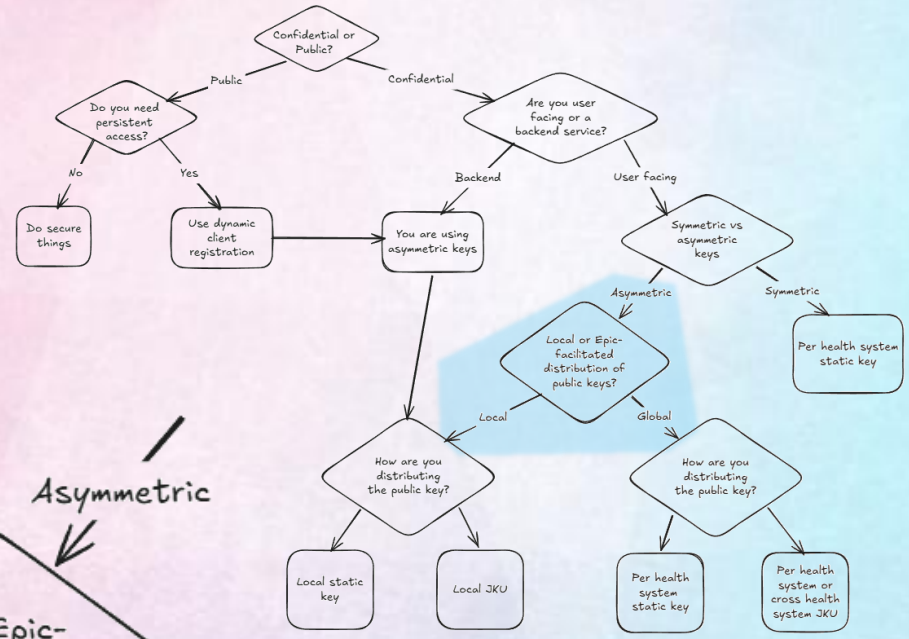


One Key



Access Token with Client Secret

Local or Global Distribution



Direct to Health System (Local)

Public Key



Asymmetric

Local or Epic-facilitated distribution of public keys?

Local

Global

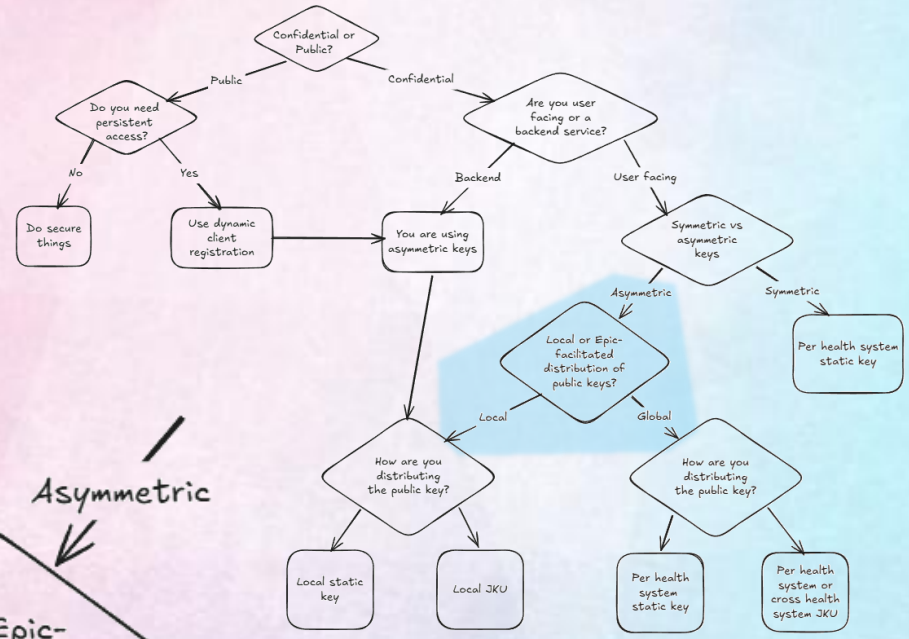
Epic-Distributed (Global)

Public Key

Epic Client Registration Site



Local or Global Distribution



Direct to Health System (Local)

Public Key



Asymmetric

Local or Epic-facilitated distribution of public keys?

Local

Global

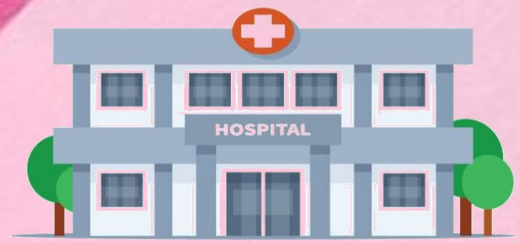
Epic-Distributed (Global)

Public Key

Epic Client Registration Site



Global Distribution of Static Keys



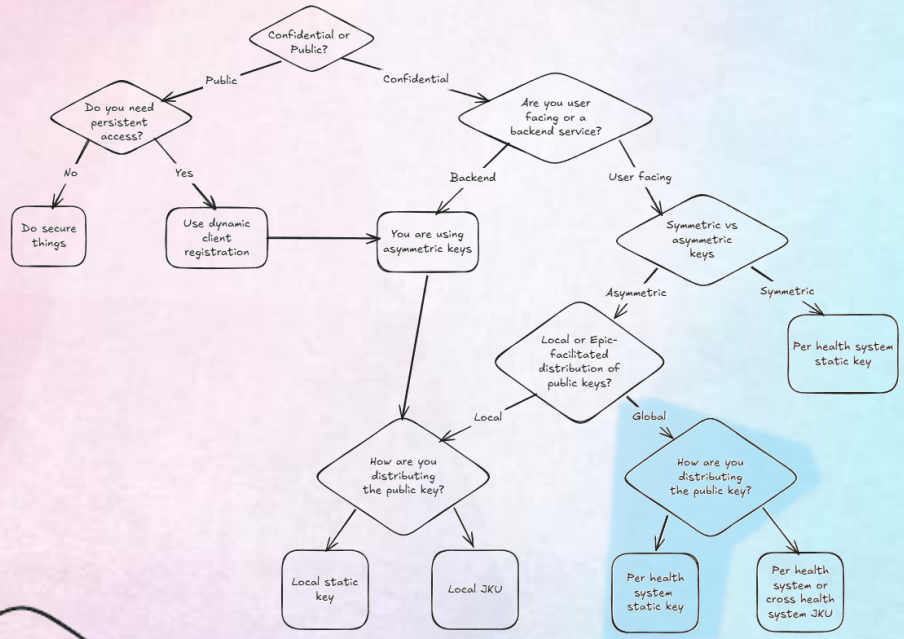
One Key

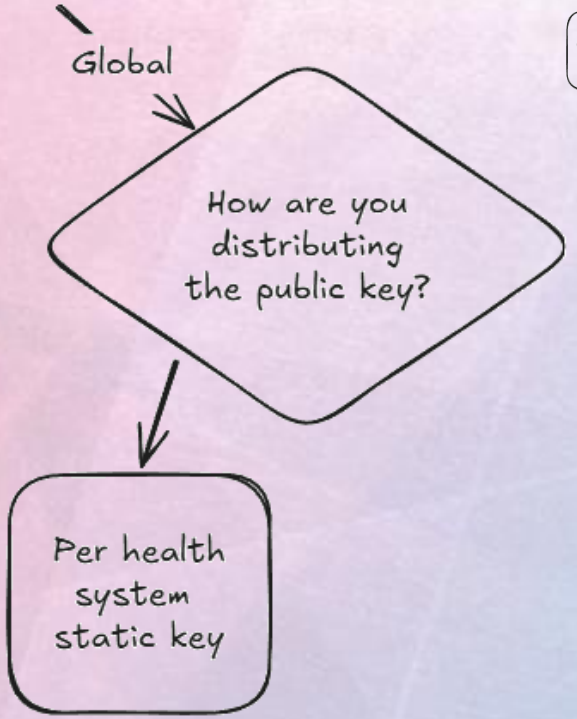


One Key

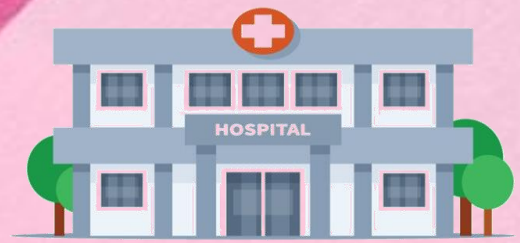


One Key





Global Distribution of Static Keys



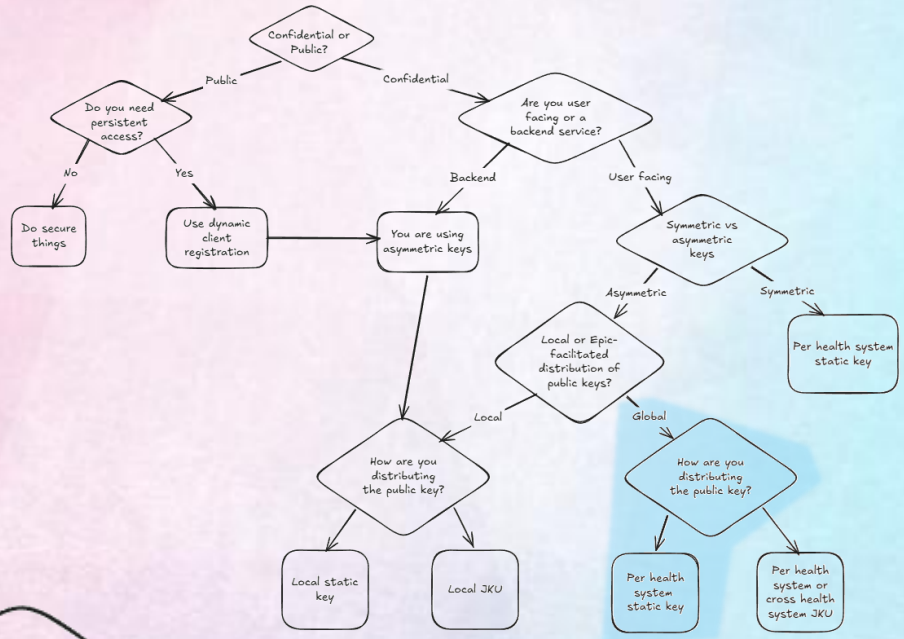
One Key

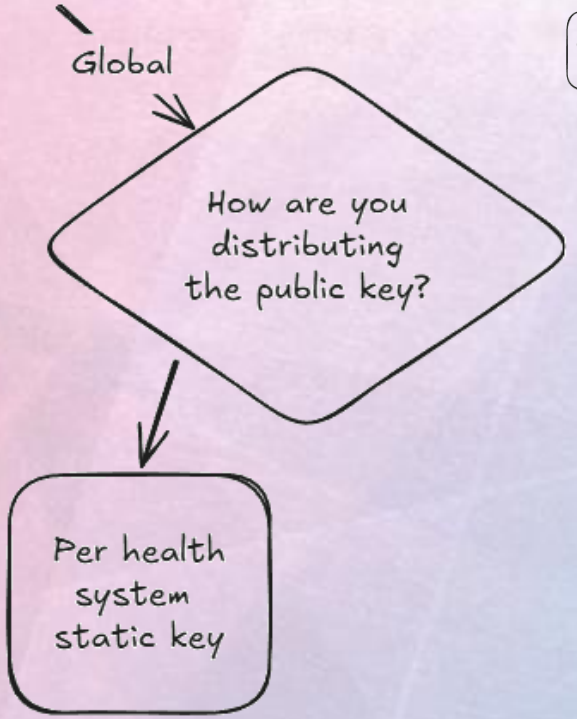


One Key

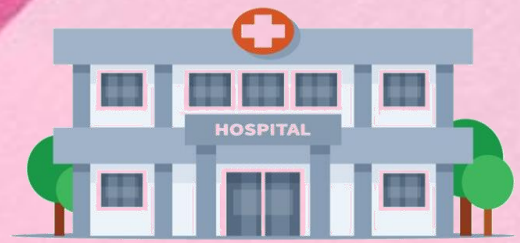


One Key





Global Distribution of Static Keys



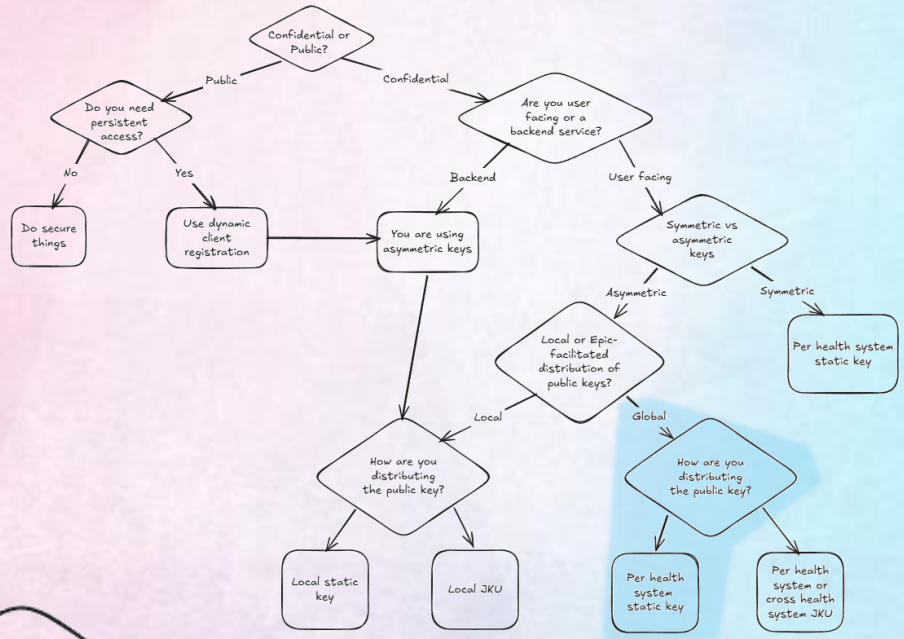
One Key

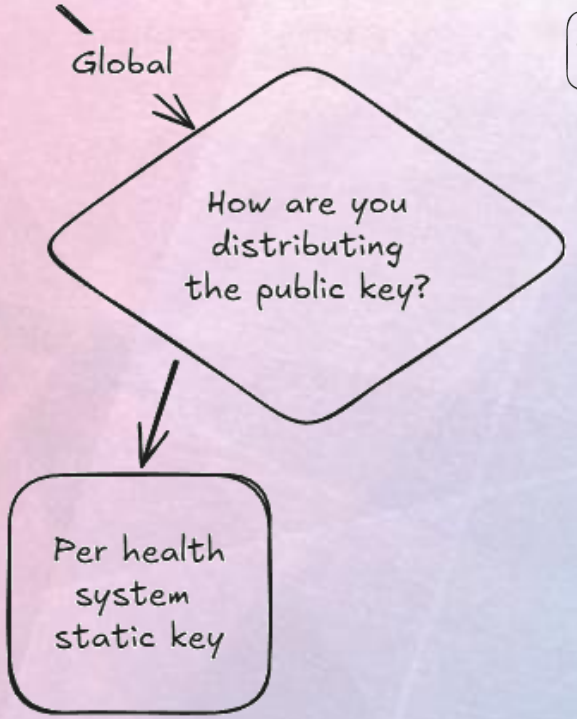


One Key

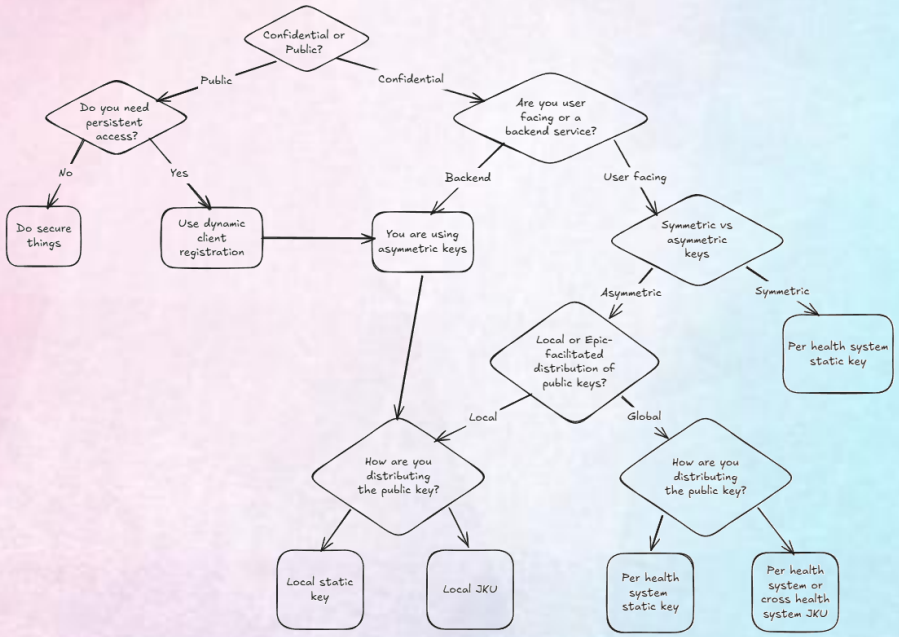


One Key





JSON Web Key Set URLs (JKUs)



Streamlined key rotation

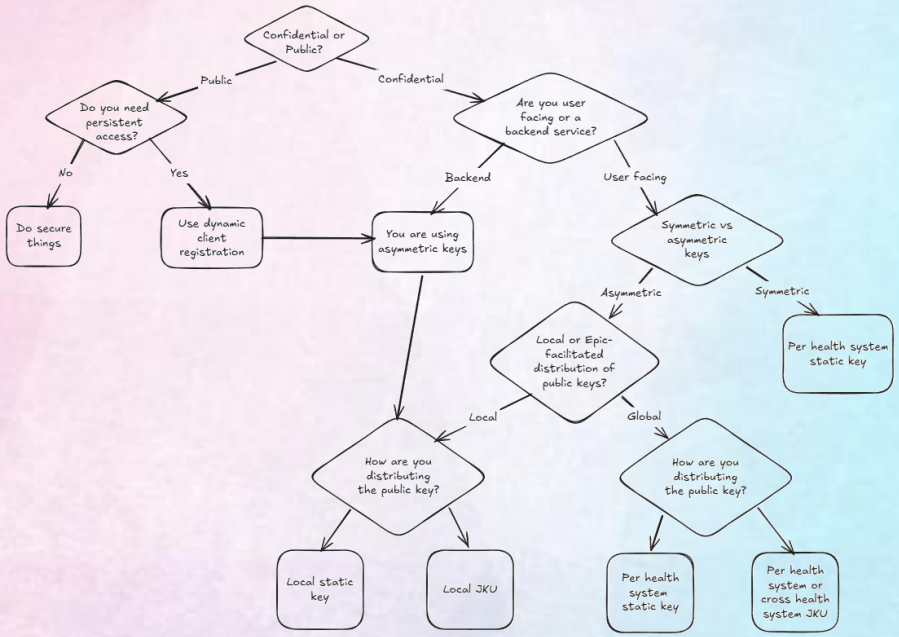
Ease of signing algorithm transitions

Simplified architecture

Decentralized architecture

Hosting a JWK Set URL

JSON Web Key Set URLs (JKUs)



Streamlined key rotation

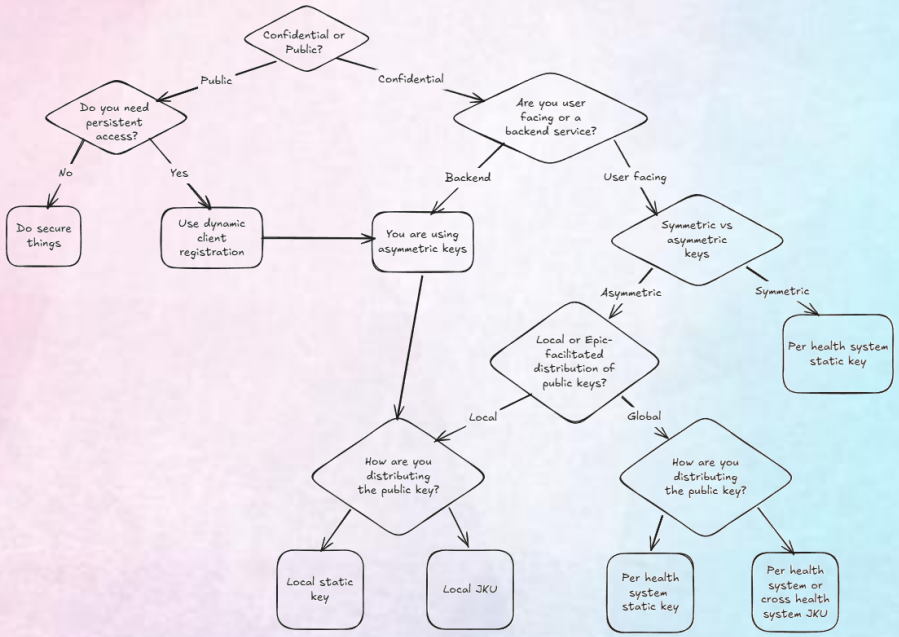
Ease of signing algorithm transitions

Simplified architecture

Decentralized architecture

Hosting a JWK Set URL

JSON Web Key Set URLs (JKUs)



Streamlined key rotation

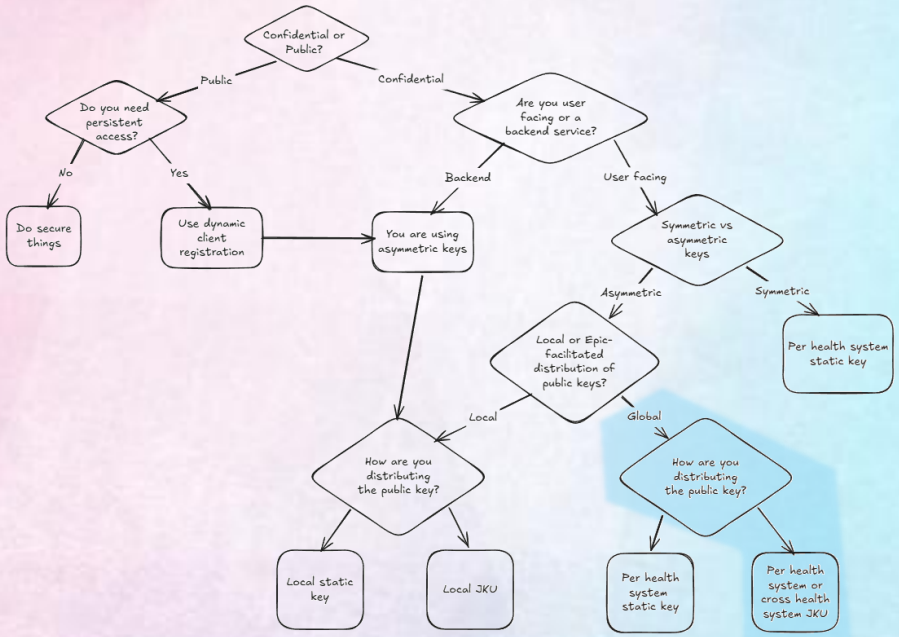
Ease of signing algorithm transitions

Simplified architecture

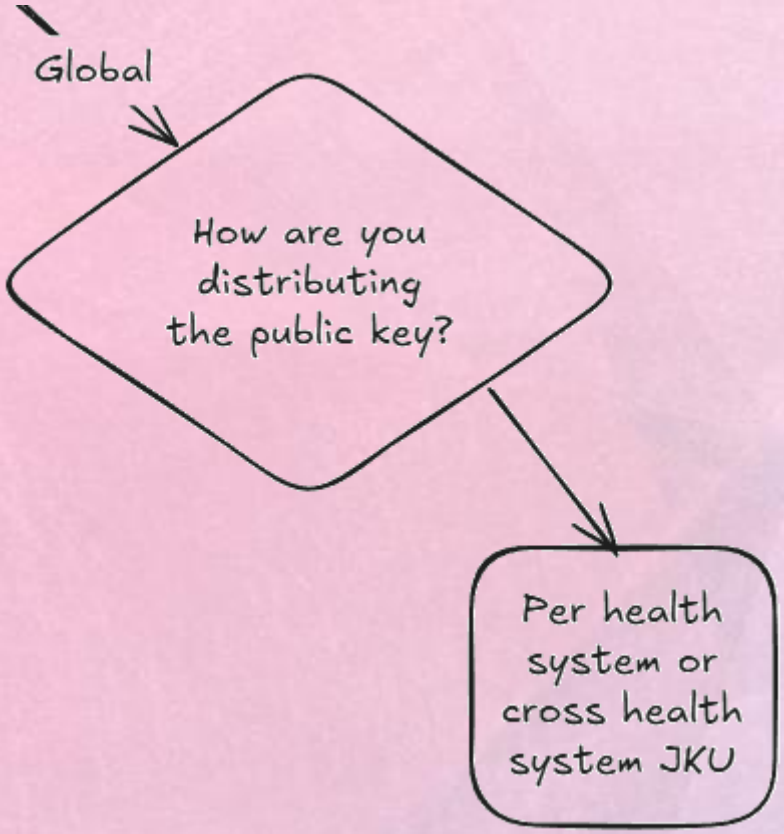
Decentralized architecture

Hosting a JWK Set URL

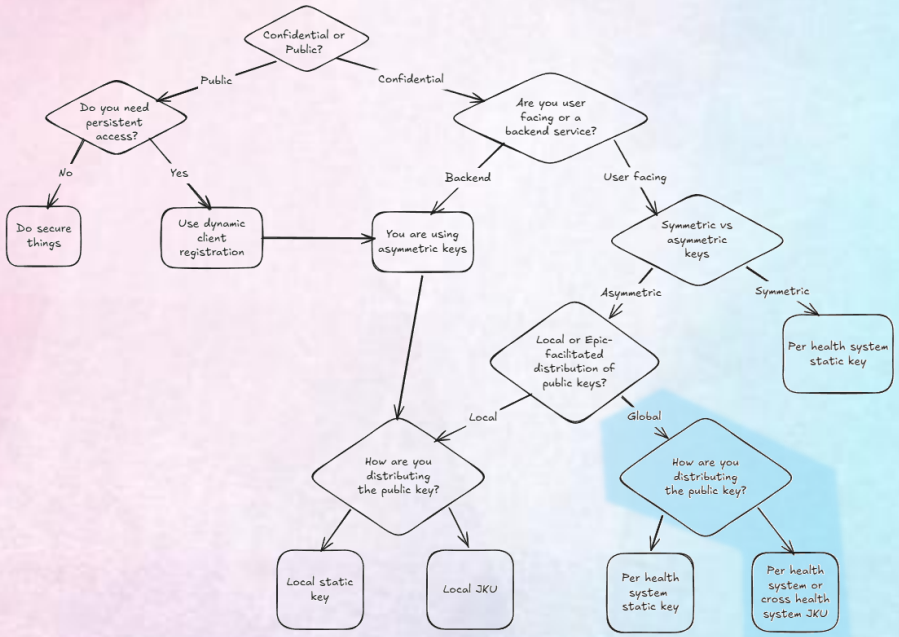
Global JKU



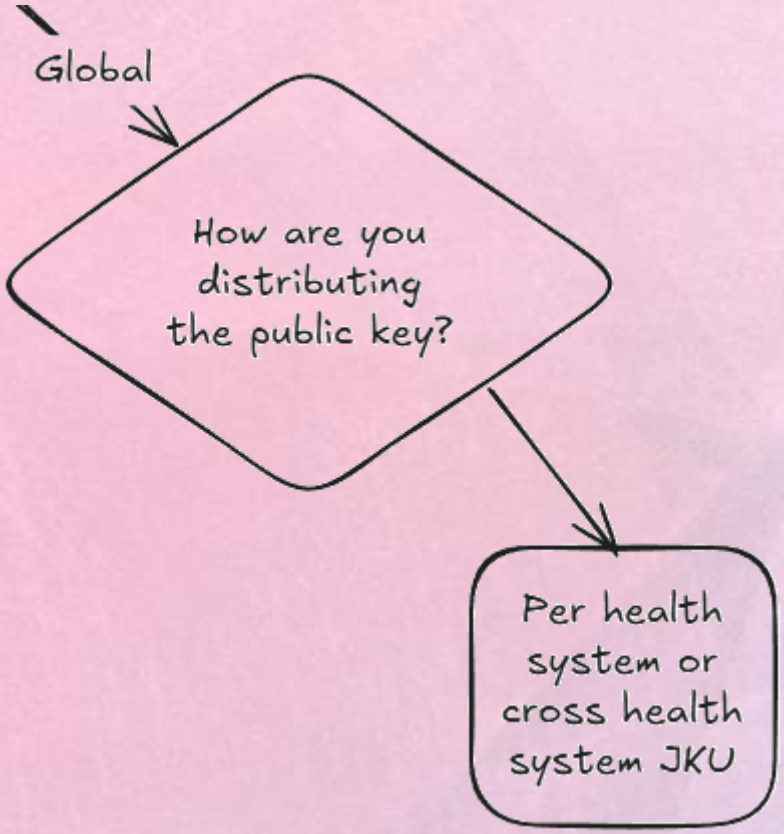
Epic-Distributed (Global)



Global JKU



Epic-Distributed (Global)



Local Keys

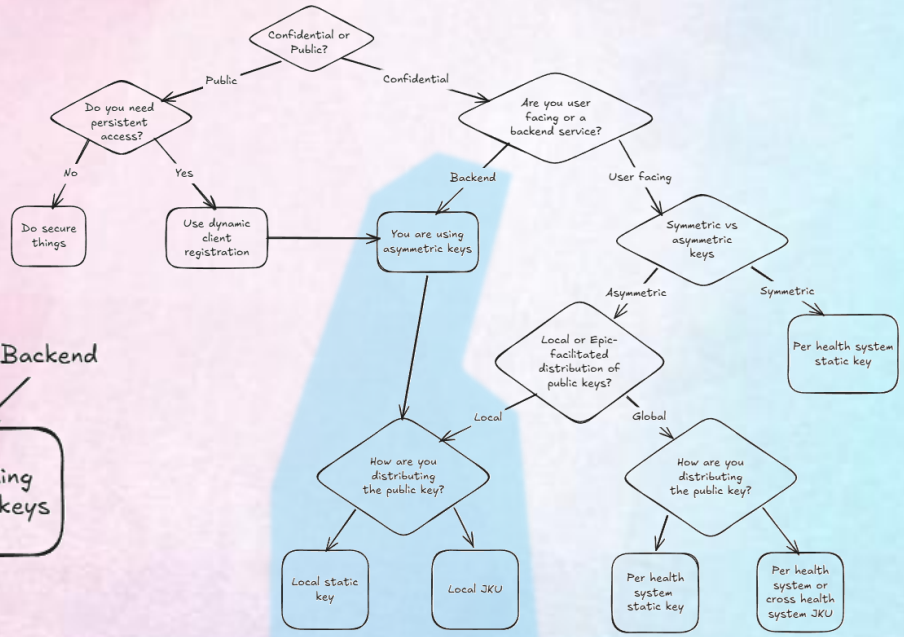
Direct to Health System (Local)

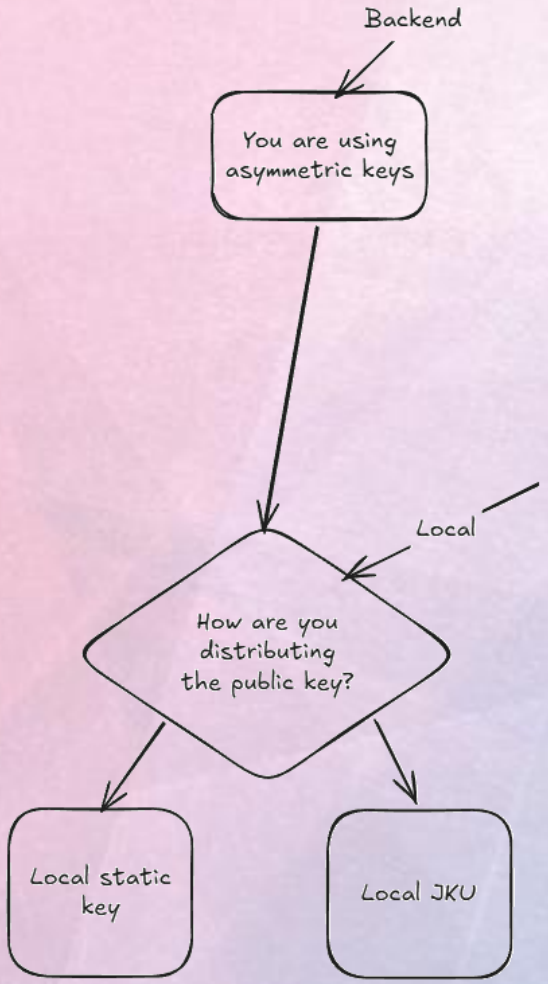
You



Health System







Local Keys

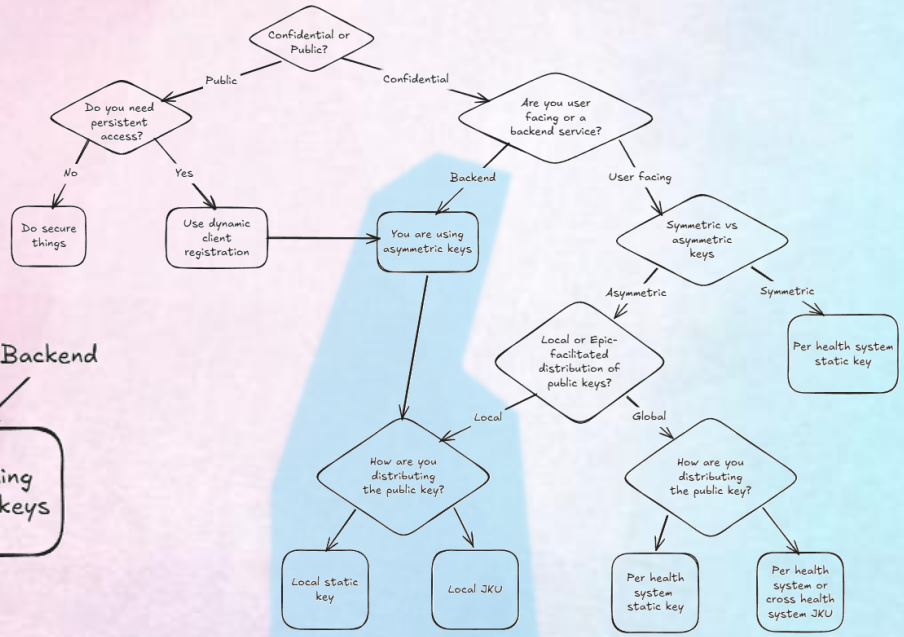
Direct to Health System (Local)

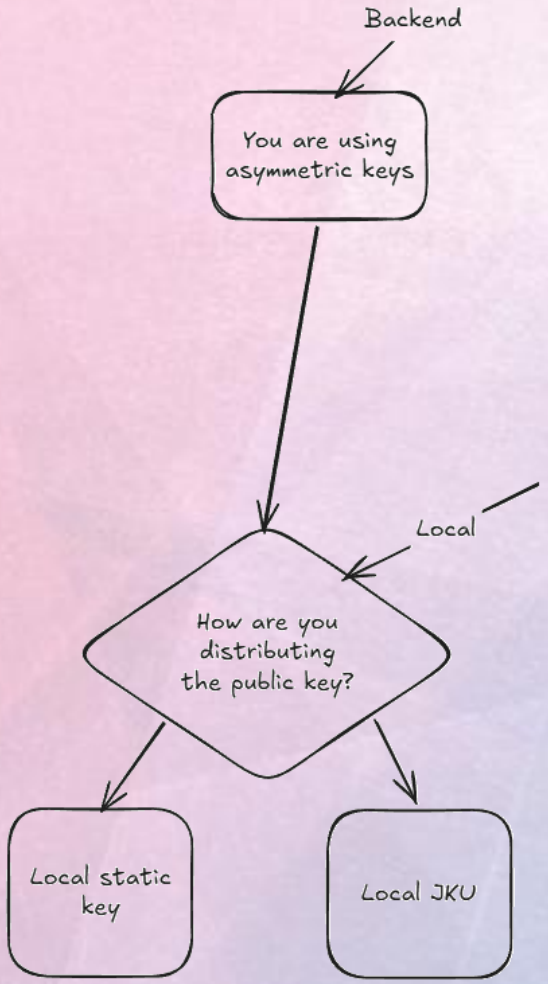
You



Health System







Local Keys

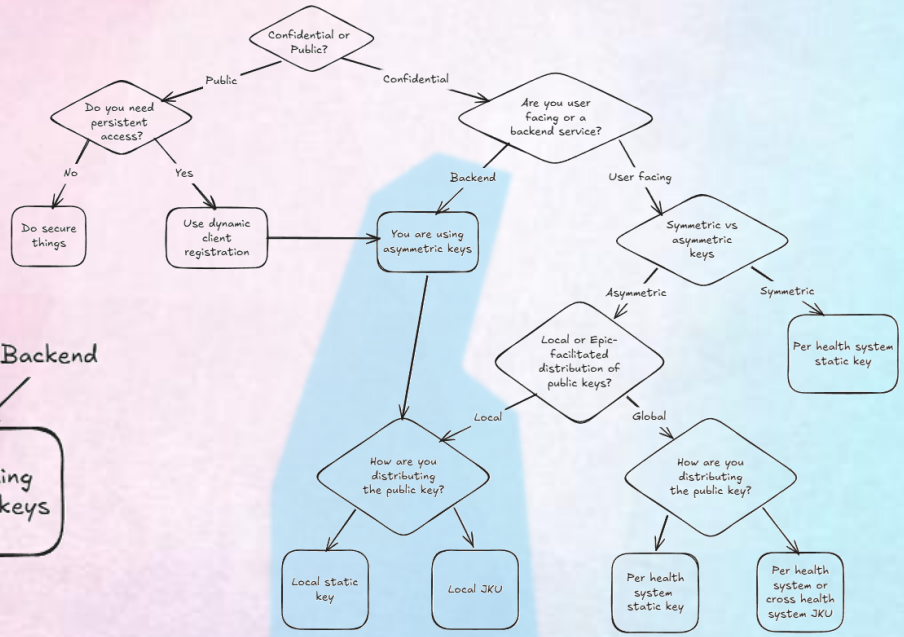
Direct to Health System (Local)

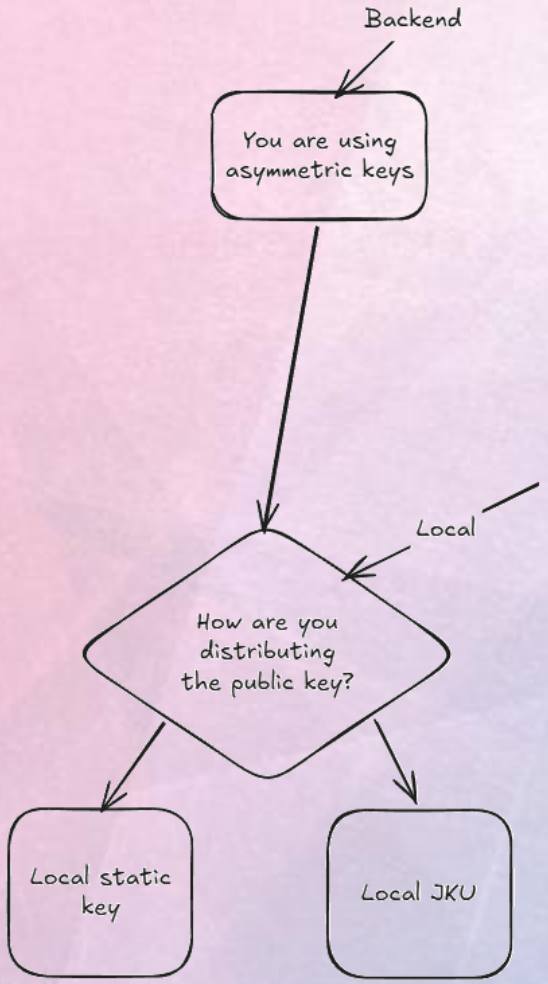
You



Health System







Seamless Experience



Persistent Access





Refresh Tokens

Launch Using Refresh Tokens

Persistent Access





Refresh Tokens

Launch Using Refresh Tokens

Persistent Access





Dynamic Client Registration

Register the Dynamic Client

Persistent Access





Dynamic Client Registration

Register the Dynamic Client

Persistent Access





Dynamic Client Registration

Register the Dynamic Client

Persistent Access





Rolling
Refresh
Tokens

Rolling Refresh
Persistent Access

Persistent Access



Indefinite
Access

Indefinite
Persistent Access

Persistent Access



Indefinite
Access

Indefinite
Persistent Access

Wrap Up

Key Takeaways

Key Takeaway:

Don't Reinvent the Wheel

OAuth 2.0 Tutorial

Choosing a Library

Using a SMART App Launch Library

A SMART app launch library needs to do all of the following:

- Run on your server, not the end-user's device.
- Support the EHR or standalone launch, depending on the launch type you chose above.
- Support private_key_jwt client authentication. We recommend this over client_secret_basic or client_secret_post.
- Support retrieving extra values from the token response. Use this option if you require a launch context beyond the standard patient and encounter identifiers.

Using an OAuth 2.0 Library

If you cannot find a SMART app launch library, here are OAuth 2.0 library requirements:

- Run on your server, not the end-user's device.
- Use an OAuth 2.0 Client Library or an OpenID Connect Relying Party library. Client refers to your role in the OAuth 2.0 flow, not whether you have a web server component.
- Support the authorization_code grant type.
- Support refresh_token grant type if you need >1 hour of API access.
- Support adding extra parameters to an authorize request ("aud" and "launch" parameters)
- Support private_key_jwt client authentication. We recommend this over client_secret_basic or client_secret_post.
- Support retrieving extra values from the token response so that you can retrieve any launch context including the standard patient and encounter identifiers.

Key Takeaway:

Review the Specs



RFC 6749: OAuth 2.0 Framework



RFC 7523: OAuth 2.0 JWT Profile



SMART App Launch



RFC 7591: Dynamic Client Registration



OpenID Connect Core 1.0



OAuth 2.0 Tutorial (Epic Authored)

Key Takeaway:

Review the Specs



RFC 6749: OAuth 2.0 Framework



RFC 7523: OAuth 2.0 JWT Profile



SMART App Launch



RFC 7591: Dynamic Client Registration



OpenID Connect Core 1.0



OAuth 2.0 Tutorial (Epic Authored)

Key Takeaway:

Make the Investment

OAuth 2.0 is worth the investment.

– Someone who has implemented OAuth 2.0



Q&A

OEC32

Wrap-Up: Putting It All Together

Form closes today at 5:00 PM

© 2026 Epic Systems Corporation. All rights reserved. PROPRIETARY INFORMATION - This item and its contents may not be accessed, used, modified, reproduced, performed, displayed, distributed or disclosed unless and only to the extent expressly authorized by an agreement with Epic. This item is a Commercial Item, as that term is defined at 48 C.F.R. Sec. 2.101. It contains trade secrets and commercial information that are confidential, privileged, and exempt from disclosure under the Freedom of Information Act and prohibited from disclosure under the Trade Secrets Act. After Visit Summary, ASAP, Aura, Beacon, Beaker, Beans, BedTime, Best Care Choices for My Patient, Bones, Break-the-Glass, Bugsy, Caboodle, Cadence, Canto, Care Everywhere, Charge Router, Cheers, Chronicles, Clarity, Cogito ergo sum, Cohort, Comfort, Community Connect, Compass Rose, Cosmos, Cosnome, Cupid, Discovery, Epic, EpicCare, EpicCare Link, Epicenter, EpicShare, EpicWeb, Epic Earth, Epic Nexus, Epic Research, Garden Plot, Grand Central, Haiku, Happy Together, Healthy Planet, Hello World, Hey Epic!, Hyperdrive, Hyperspace, Kaleidoscope, Kit, Limerick, Lucy, Lumens, MyChart, Nebula, OpTime, Phoenix, Powered by Epic, Prelude, Radar, Radiant, Resolute, Revenue Guardian, Rover, Share Everywhere, SmartForms, Sonnet, Stork, System Pulse, Tapestry, Trove, Welcome, Willow, Wisdom, With the Patient at Heart, and WorldWise are registered trademarks, trademarks, or service marks of Epic Systems Corporation in the United States of America and/or other countries. Other company, product, and service names referenced herein may be trademarks or service marks of their respective owners. Patents Notice: www.epic.com/patents.