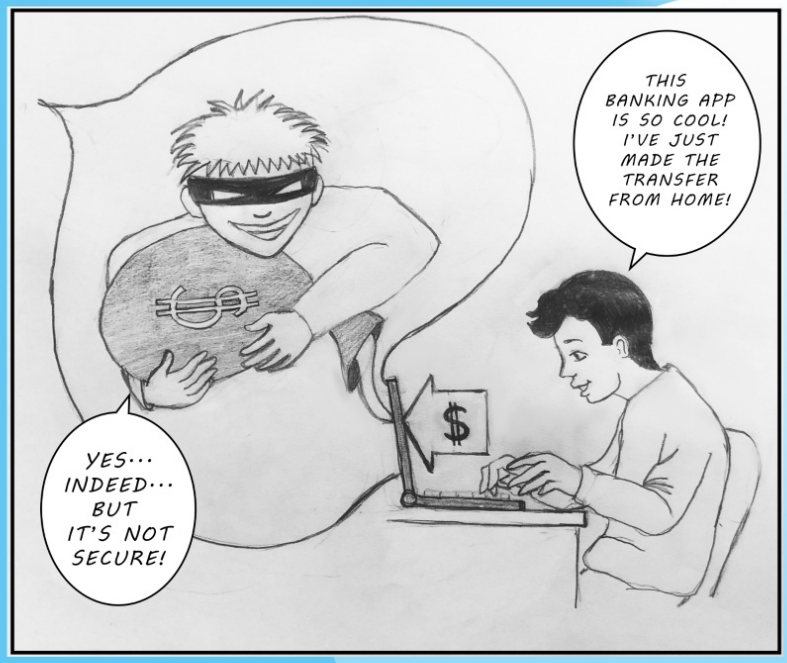
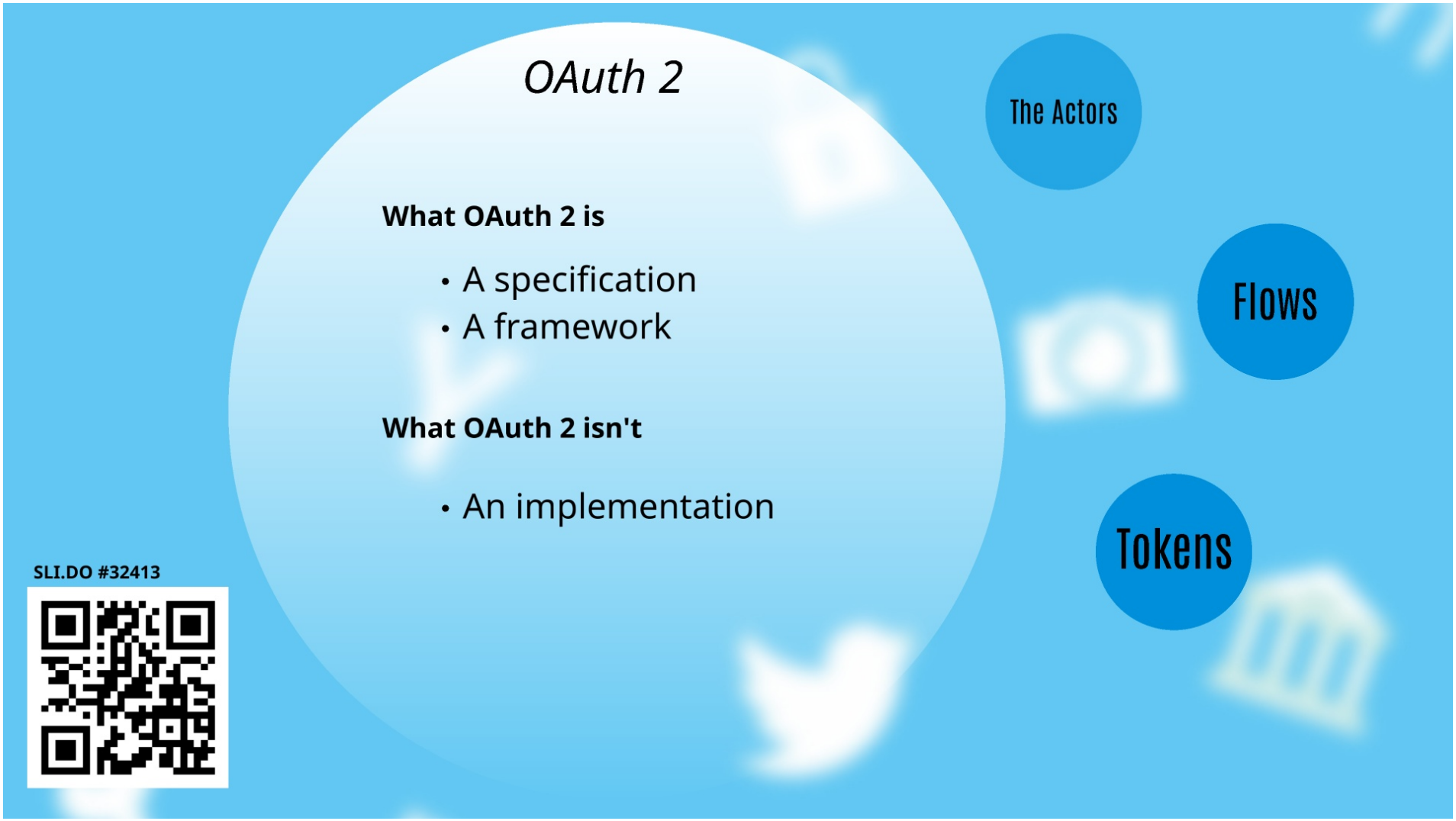

endava
How
Security
Hands-on
OAuth 2
Using OAuth 2
With Spring Security
A Masterclass
SLI.DO #32413



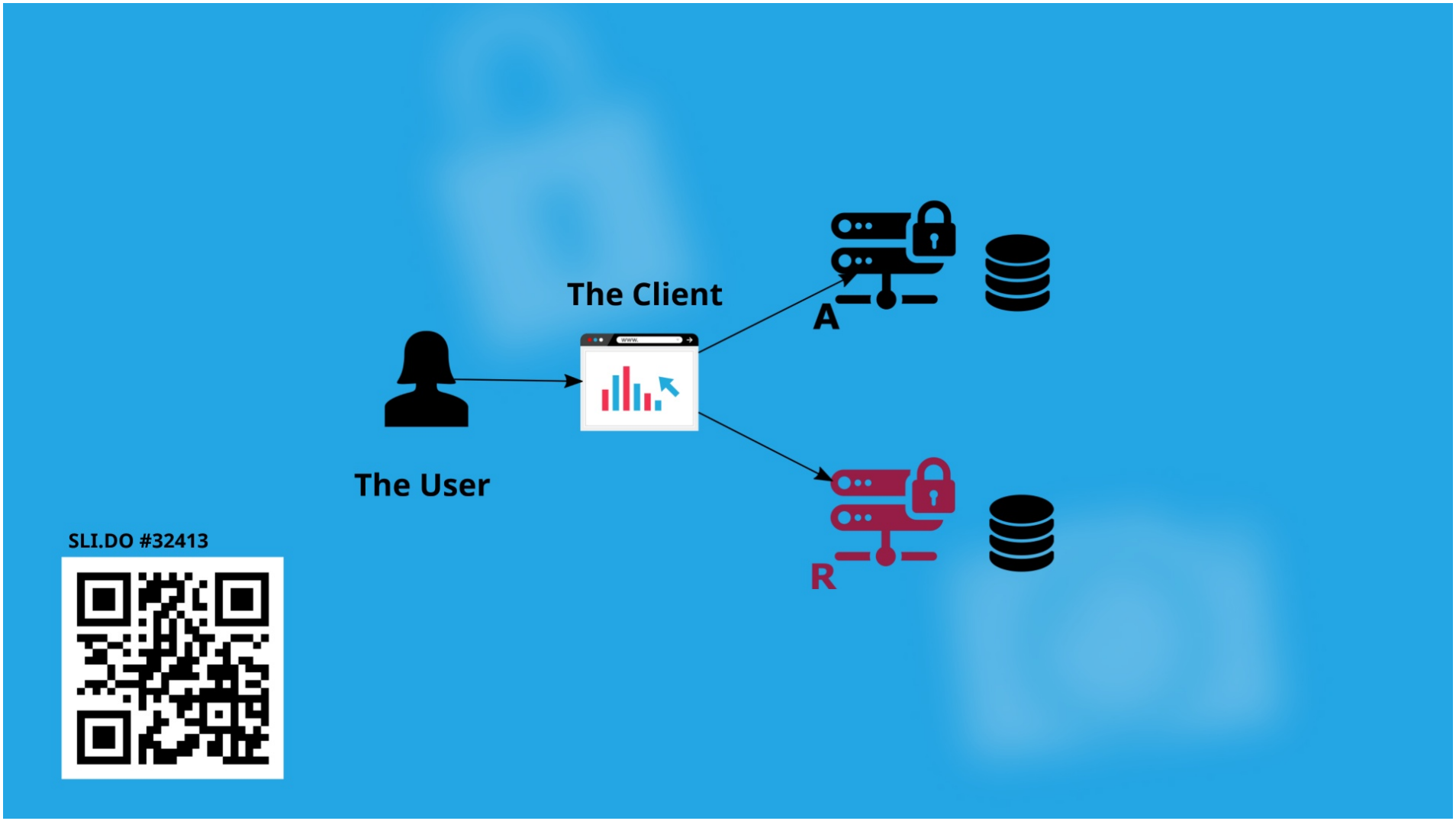
- cross-site request forgery
- session fixation
- injection
- broken authentication
- broken authorization
- unsecured data storage
- unsecured data transition

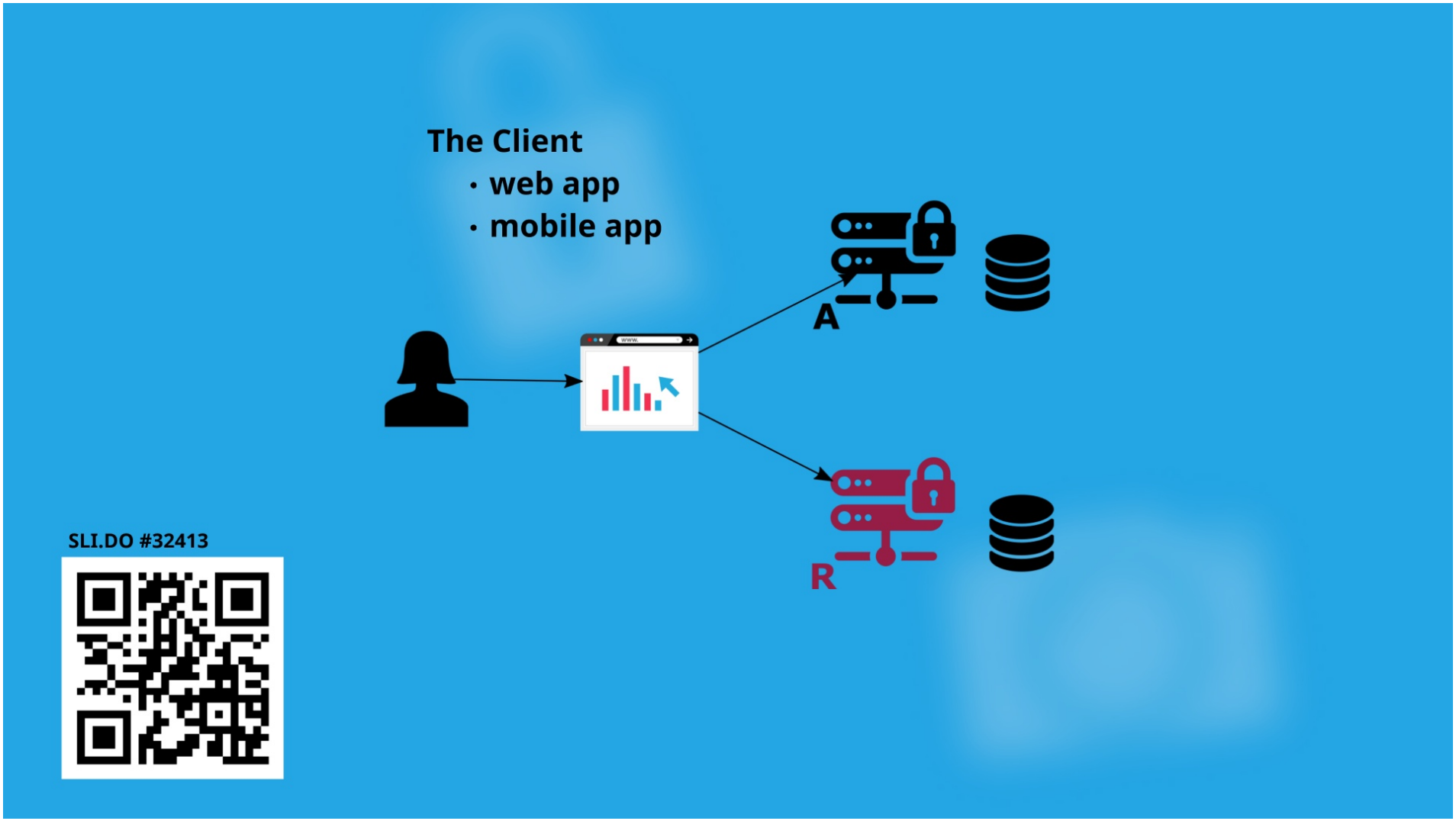
SLI.DO #32413


endava
How
Security
Hands-on
OAuth 2
Using OAuth 2
With Spring Security
A Masterclass
SLI.DO #32413


OAuth 2
What OAuth 2 is
• A specification
• A framework
What OAuth 2 isn't
• An implementation
SLI.DO #32413
The Actors
Flows
Tokens

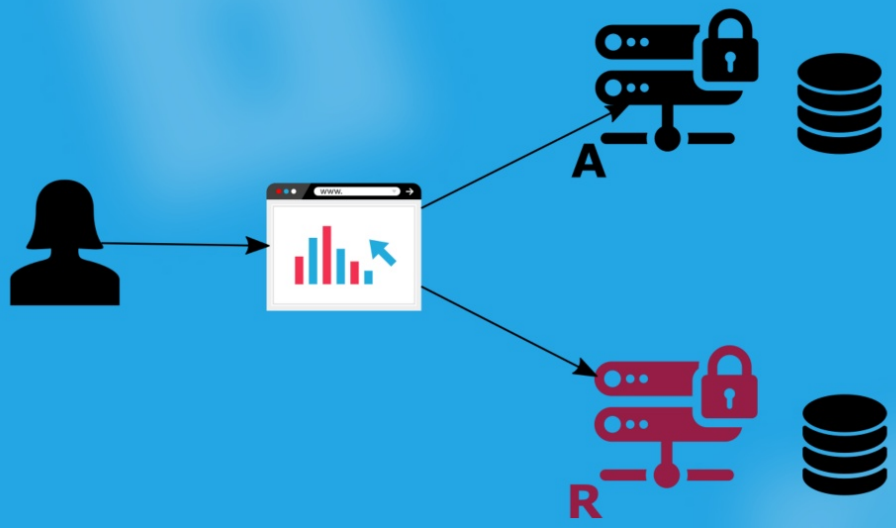

A
R
SLI.DO #32413


The Client
A
The User
R
SLI.DO #32413

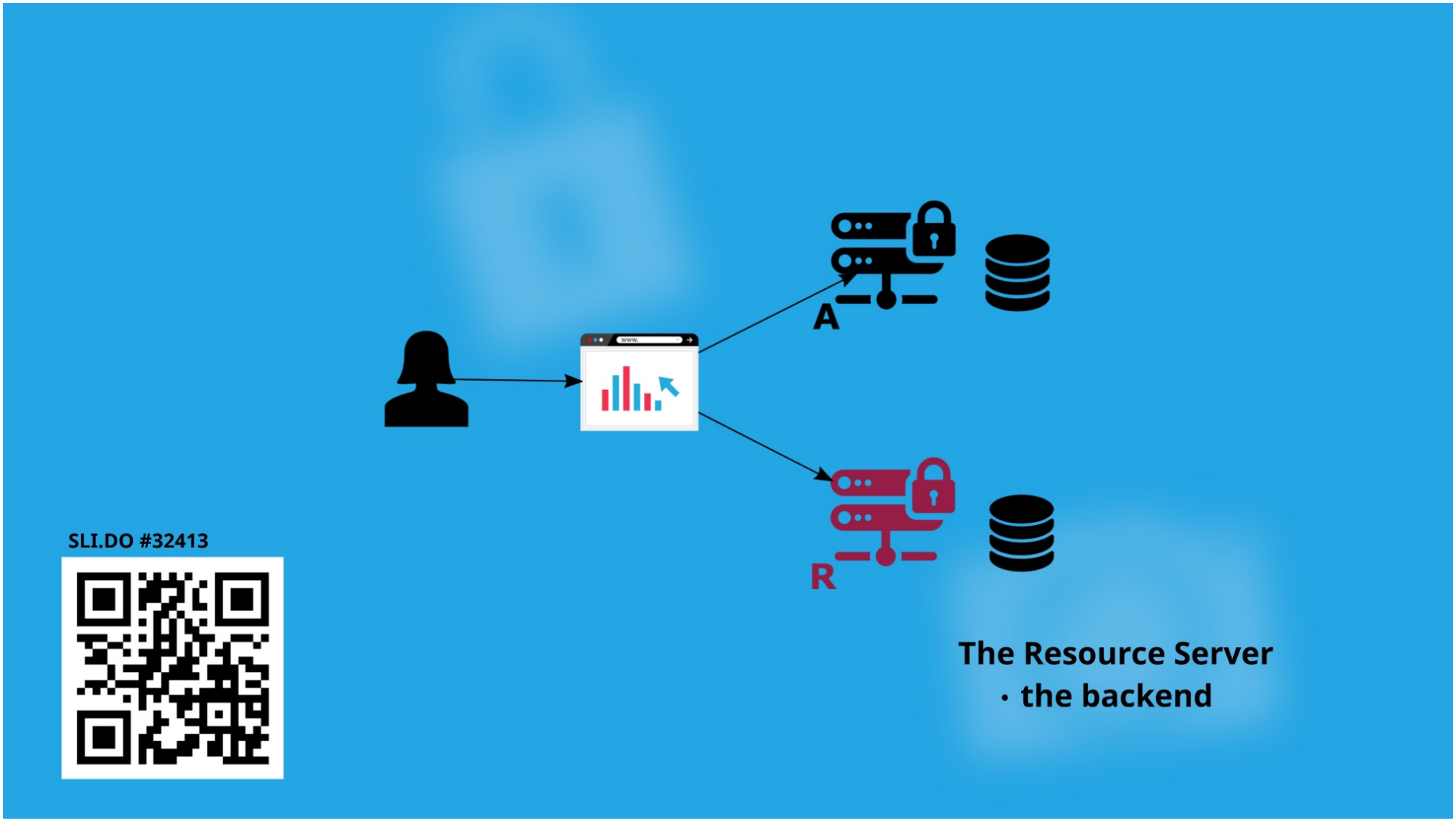

The Client
• web app
• mobile app
A
R
SLI.DO #32413

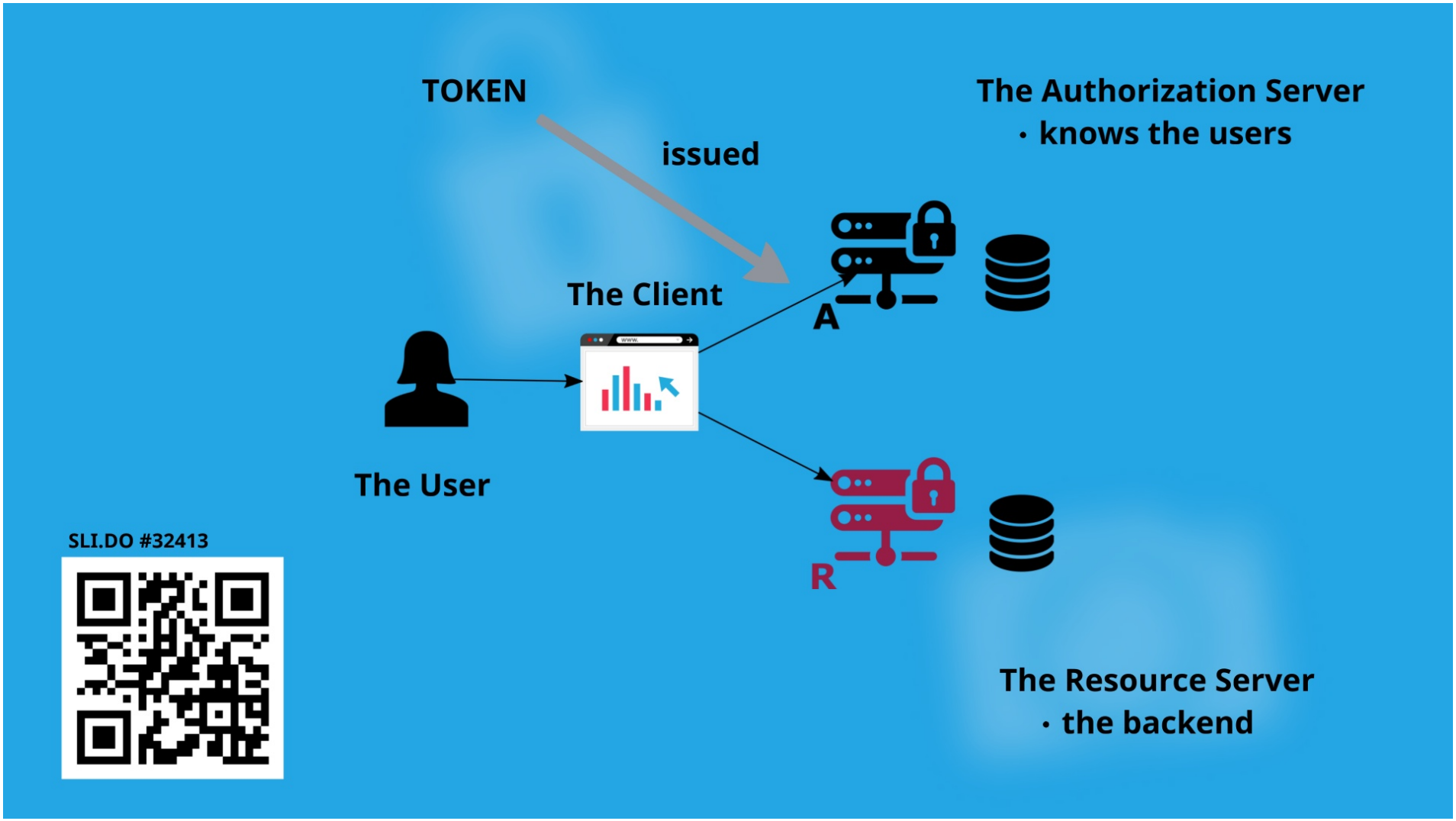
The Authorization Server

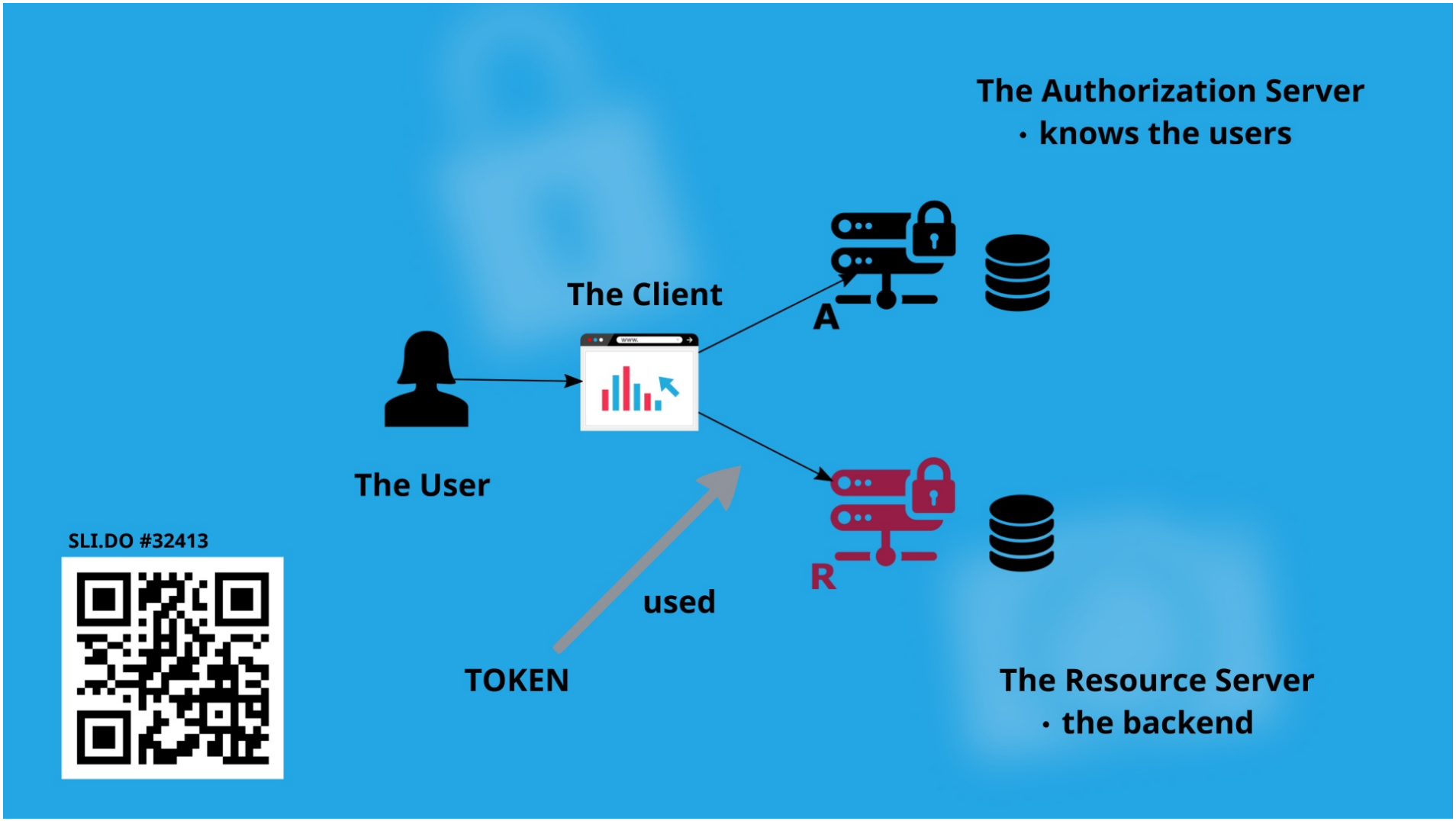
- knows the users

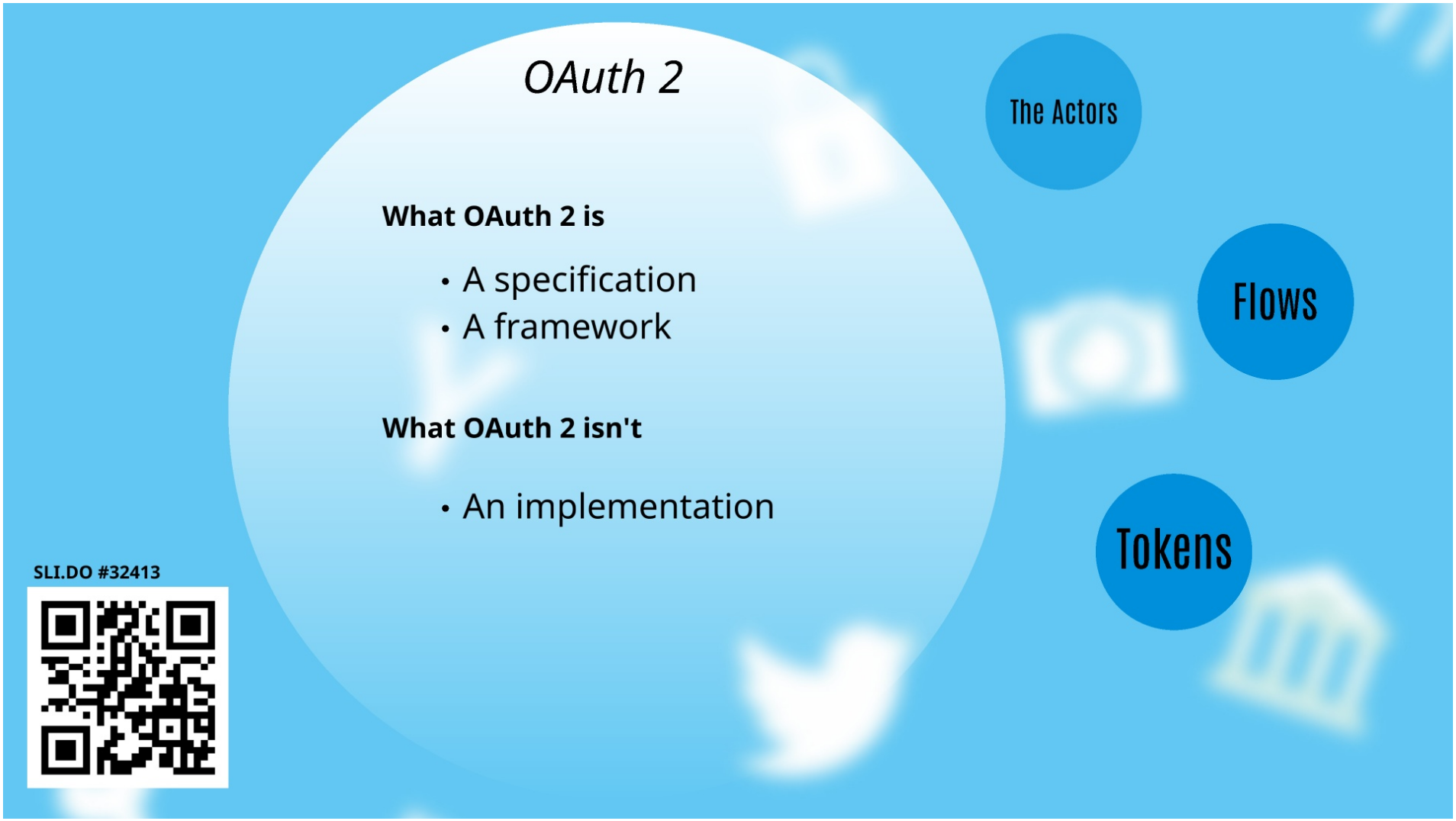


SLI.DO #32413


A
R
SLI.DO #32413
The Resource Server
· the backend


TOKEN
issued
The Authorization Server
• knows the users
The Client
A
The User
SLI.DO #32413
R
The Resource Server
• the backend


The Authorization Server
• knows the users
The Client
A
The User
SLI.DO #32413
used
R
TOKEN
The Resource Server
• the backend


OAuth 2
What OAuth 2 is
• A specification
• A framework
What OAuth 2 isn't
• An implementation
SLI.DO #32413
The Actors
Flows
Tokens

GRANT TYPES

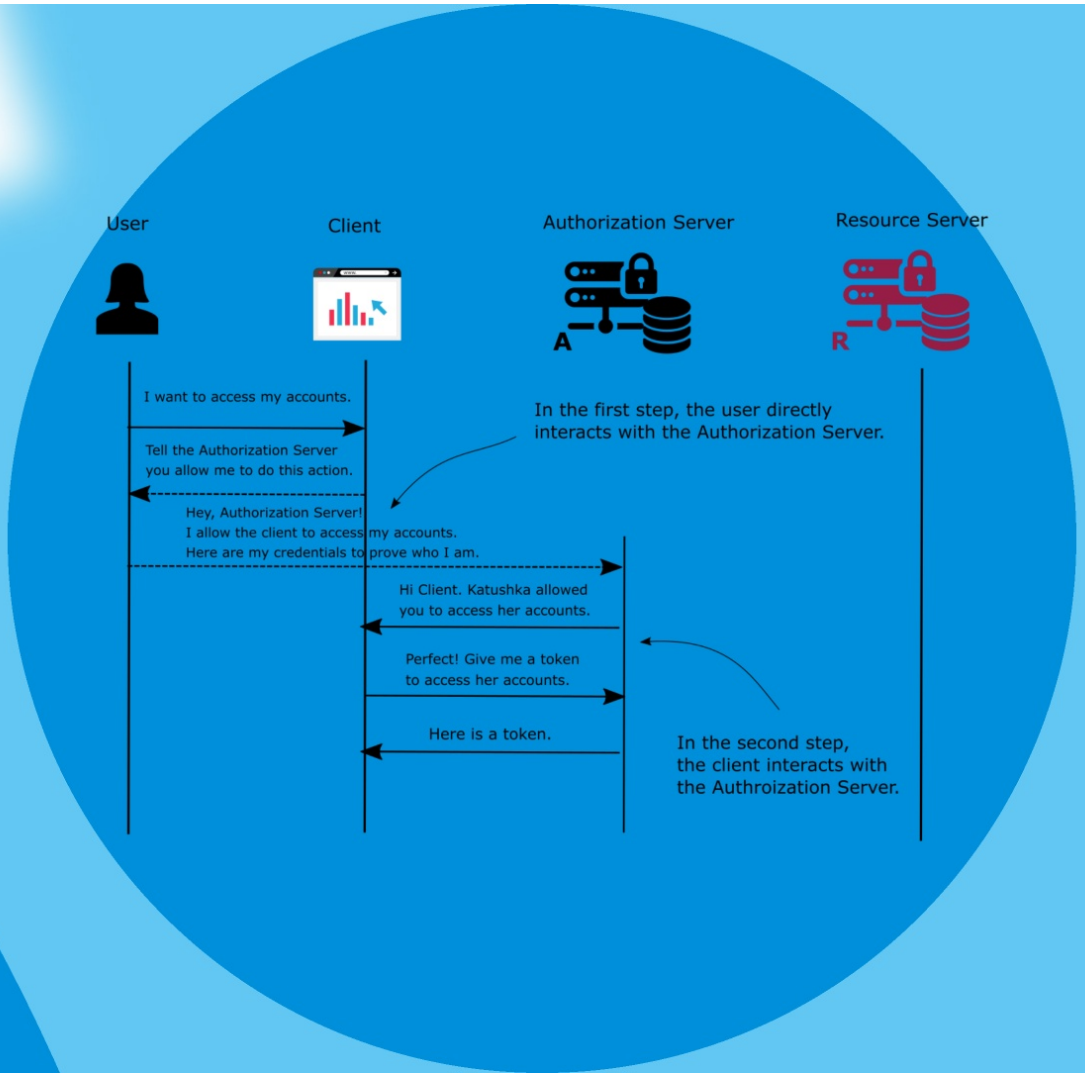
- implicit grant type (deprecated)
- authorization code grant type
- password grant type (deprecated)
- client credentials grant type

AC

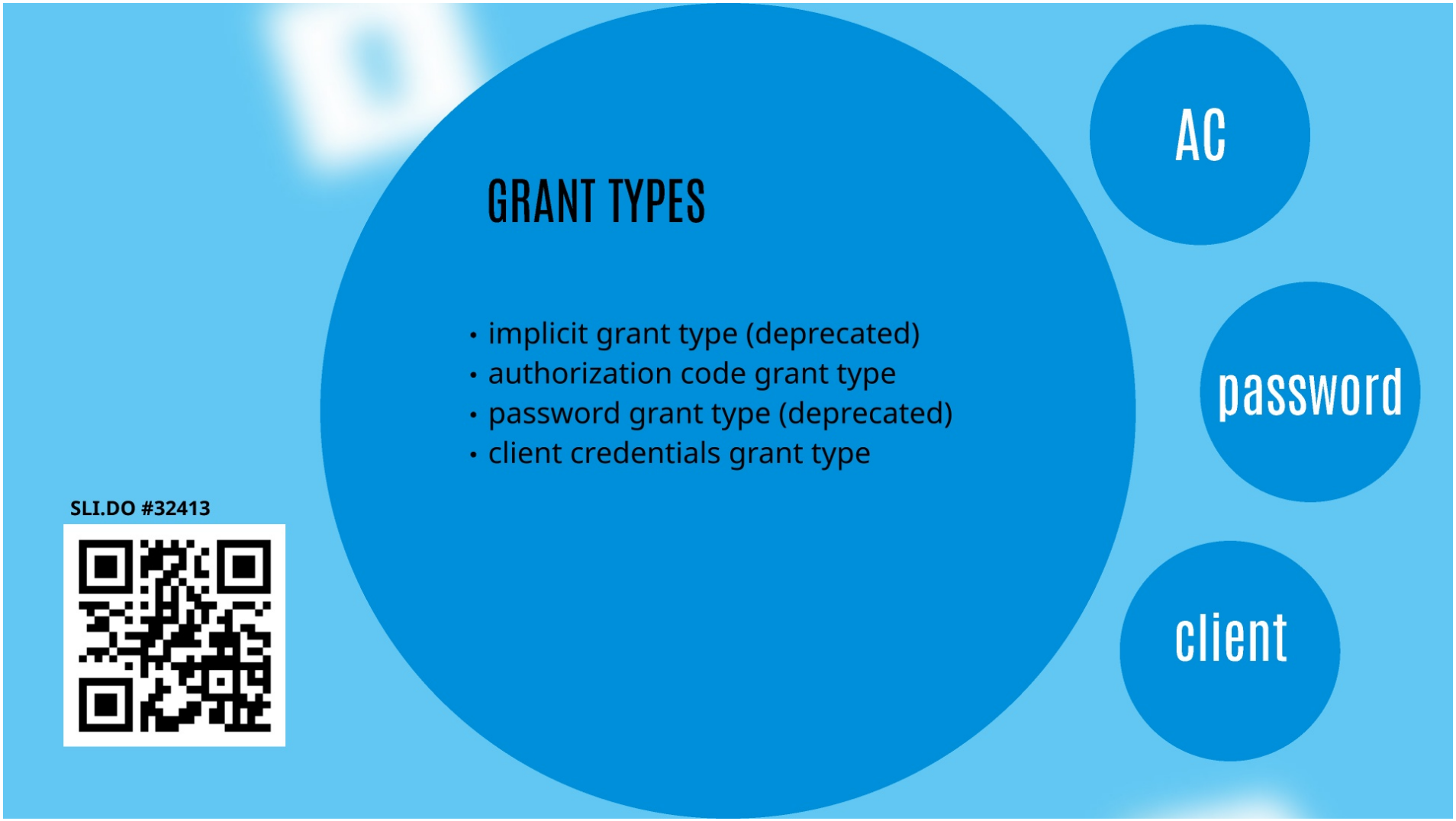
password

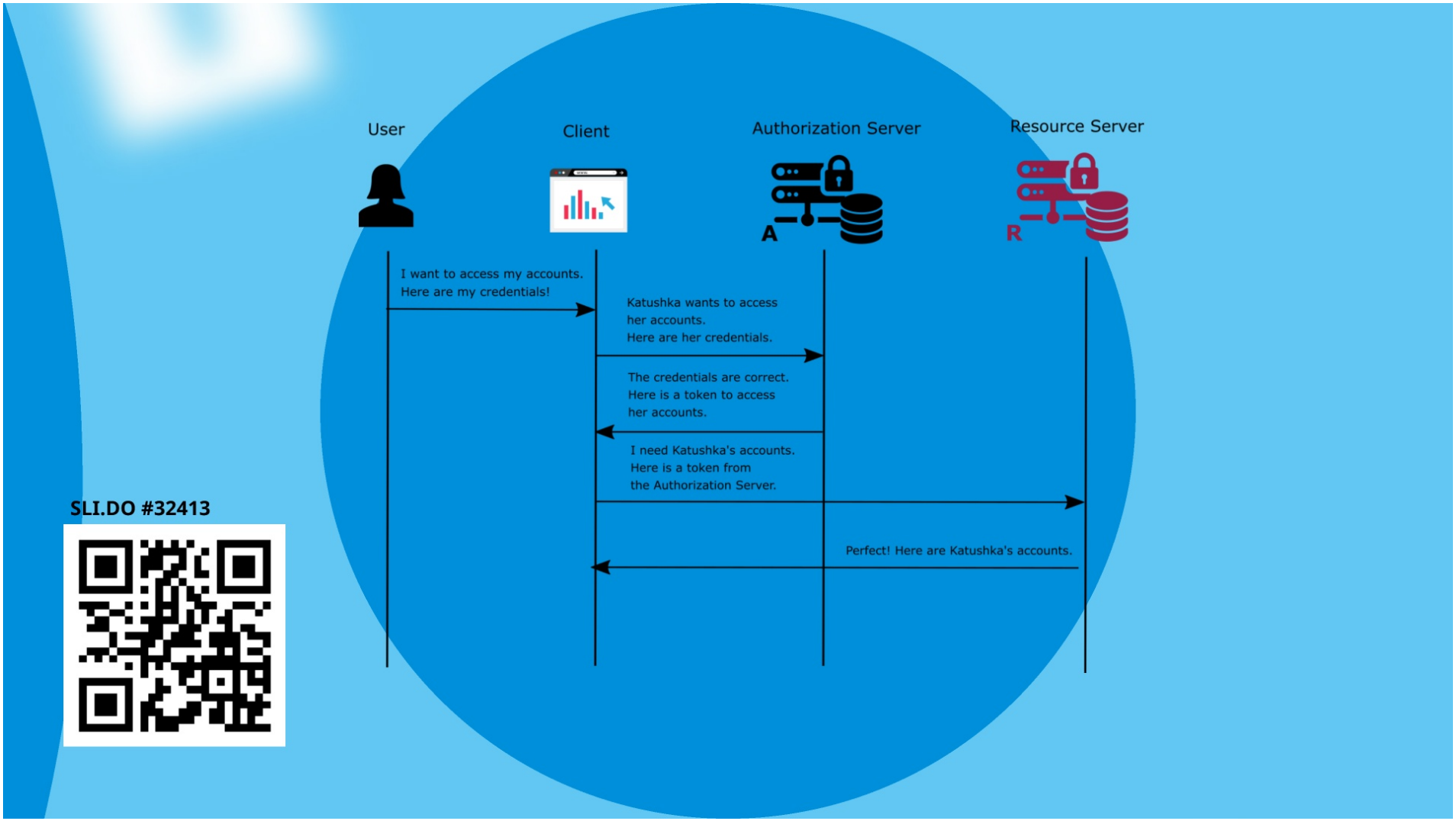
client

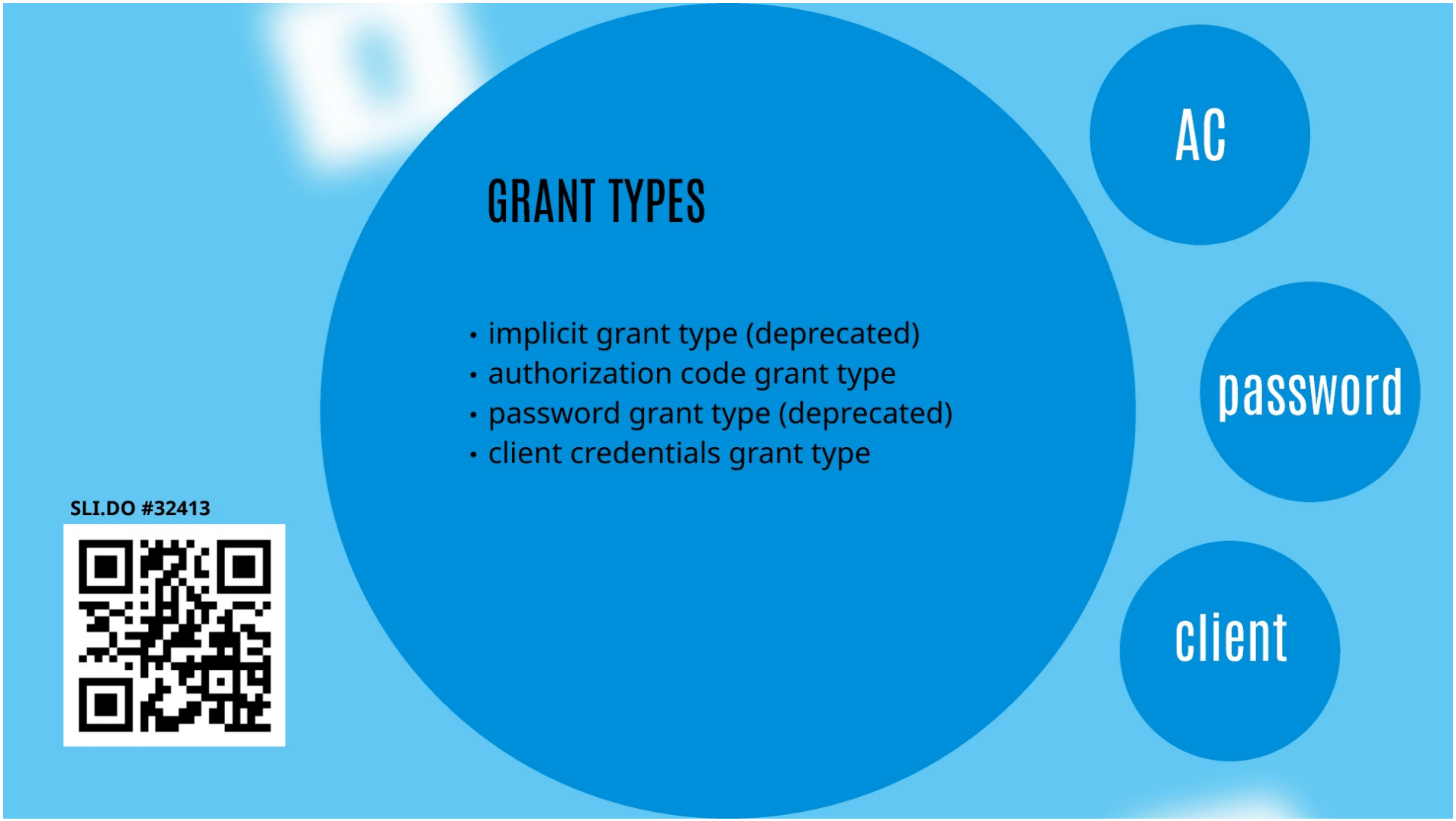
SLI.DO #32413

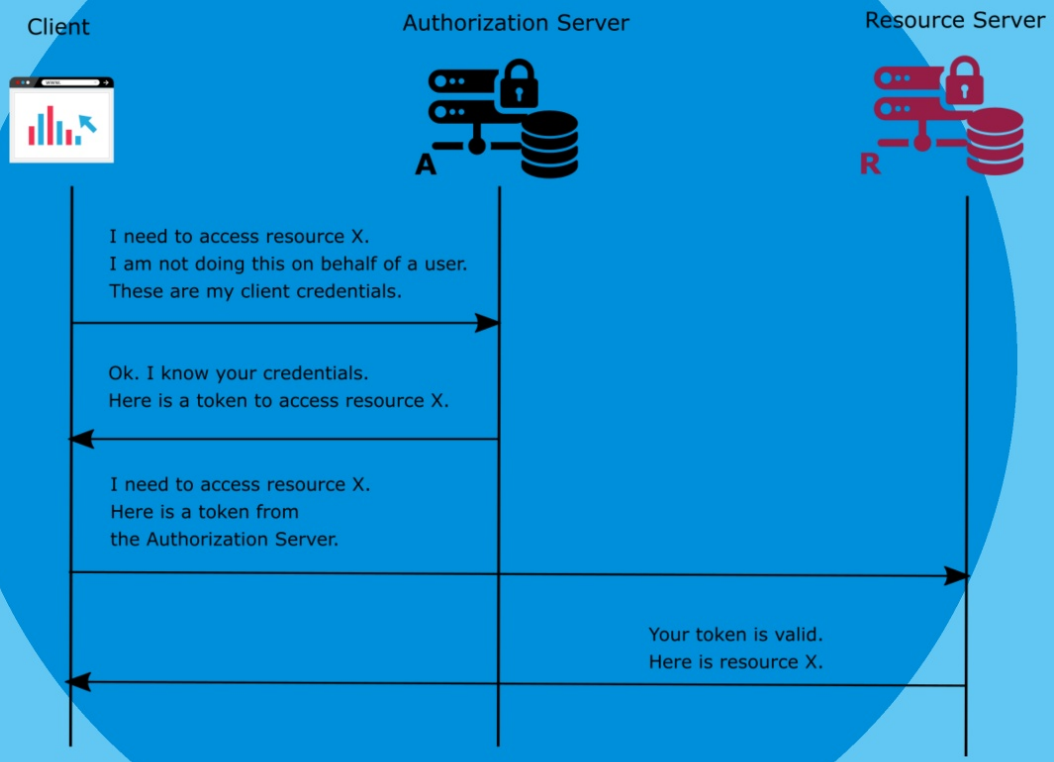

User
Client
Authorization Server
Resource Server
A
R
I want to access my accounts.
Tell the Authorization Server
you allow me to do this action.
Hey, Authorization Server!
I allow the client to access my accounts.
Here are my credentials to prove who I am.
Hi Client. Katushka allowed
you to access her accounts.
Perfect! Give me a token
to access her accounts.
Here is a token.
In the first step, the user directly
interacts with the Authorization Server.
In the second step,
the client interacts with
the Authroization Server.

SLI.DO #32413

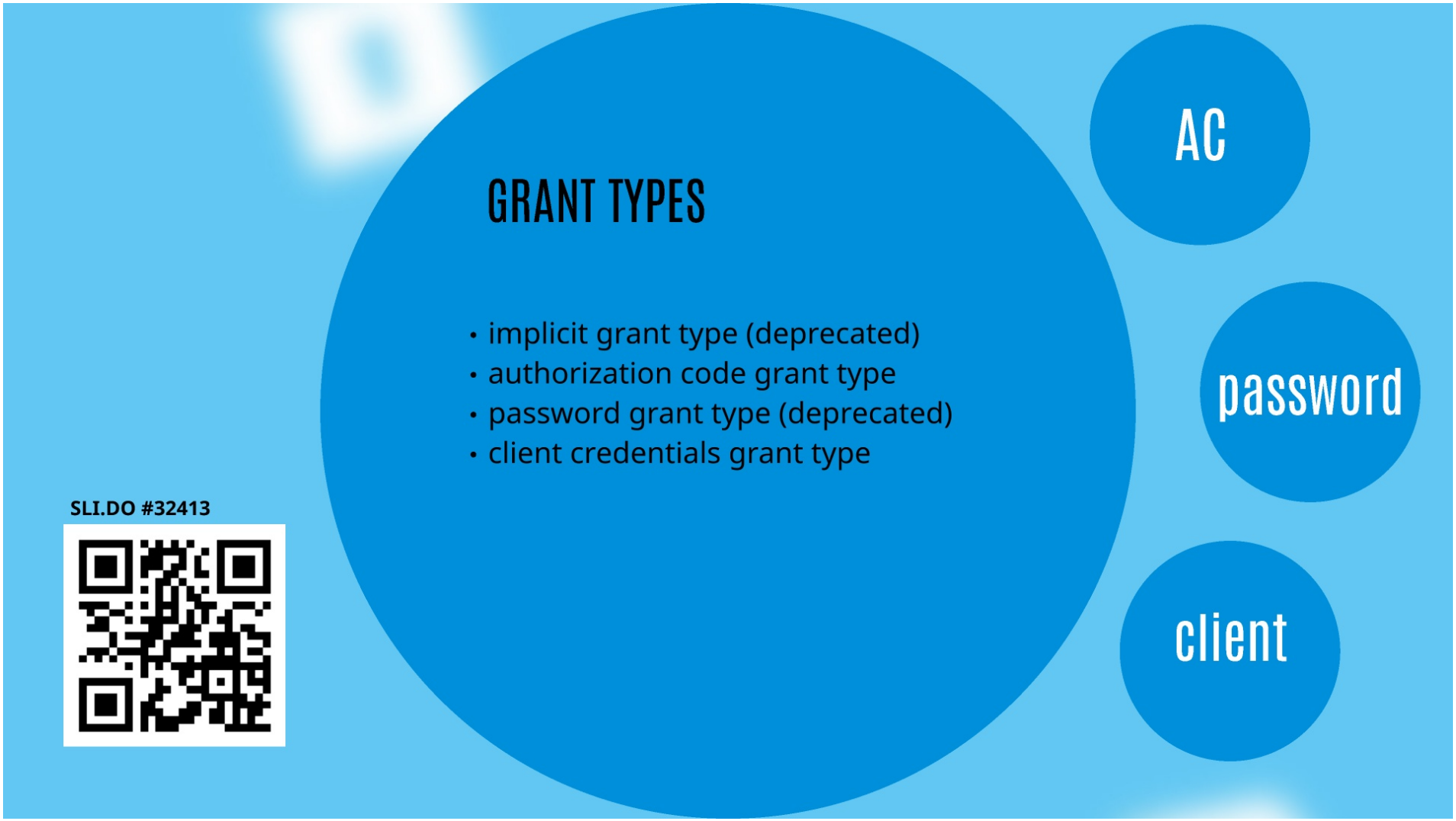

GRANT TYPES
• implicit grant type (deprecated)
• authorization code grant type
• password grant type (deprecated)
• client credentials grant type
AC
password
client
SLI.DO #32413

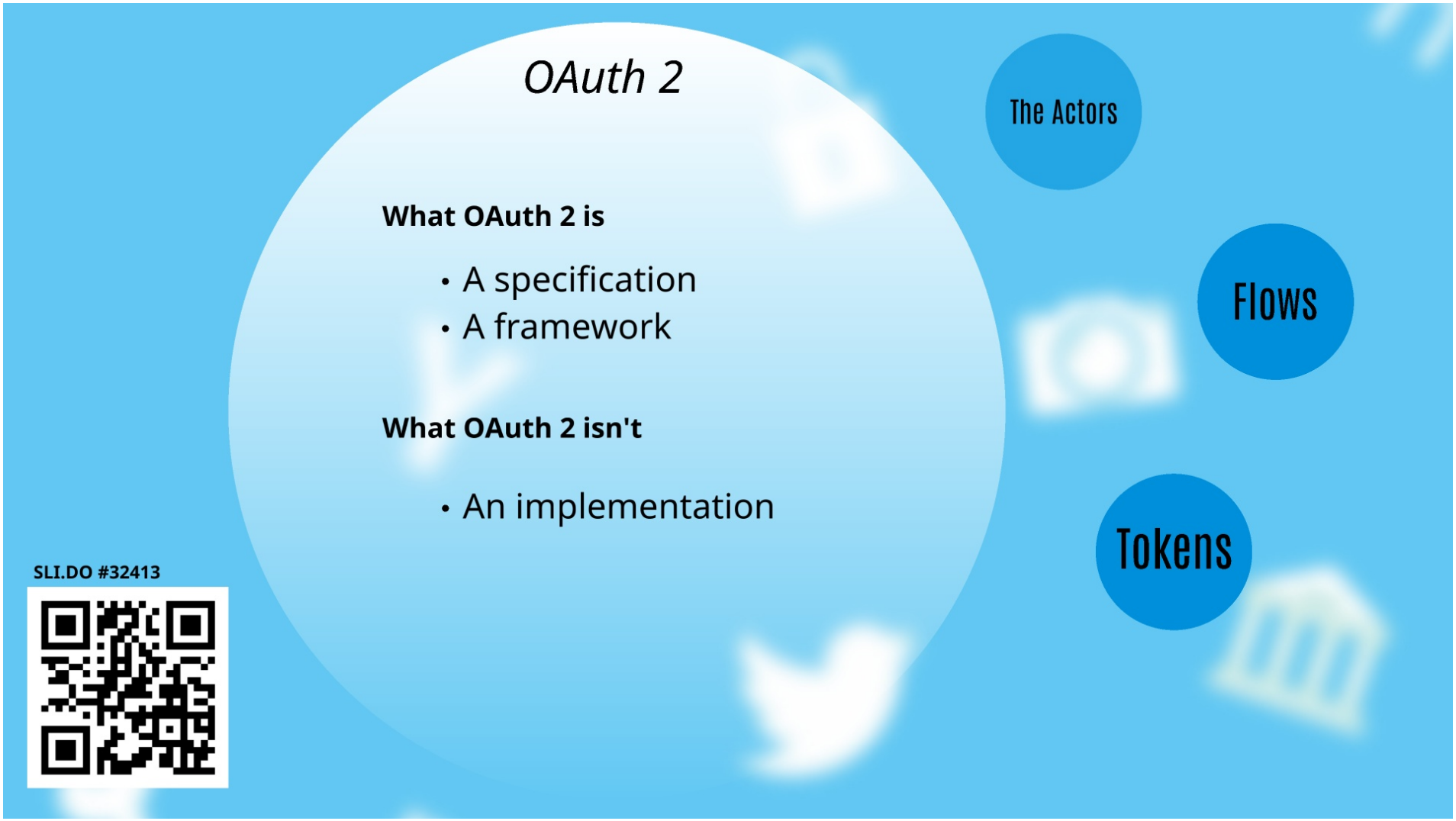

User
Client
Authorization Server
Resource Server
A
R
I want to access my accounts.
Here are my credentials!
Katushka wants to access
her accounts.
Here are her credentials.
The credentials are correct.
Here is a token to access
her accounts.
I need Katushka's accounts.
Here is a token from
the Authorization Server.
Perfect! Here are Katushka's accounts.
SLI.DO #32413

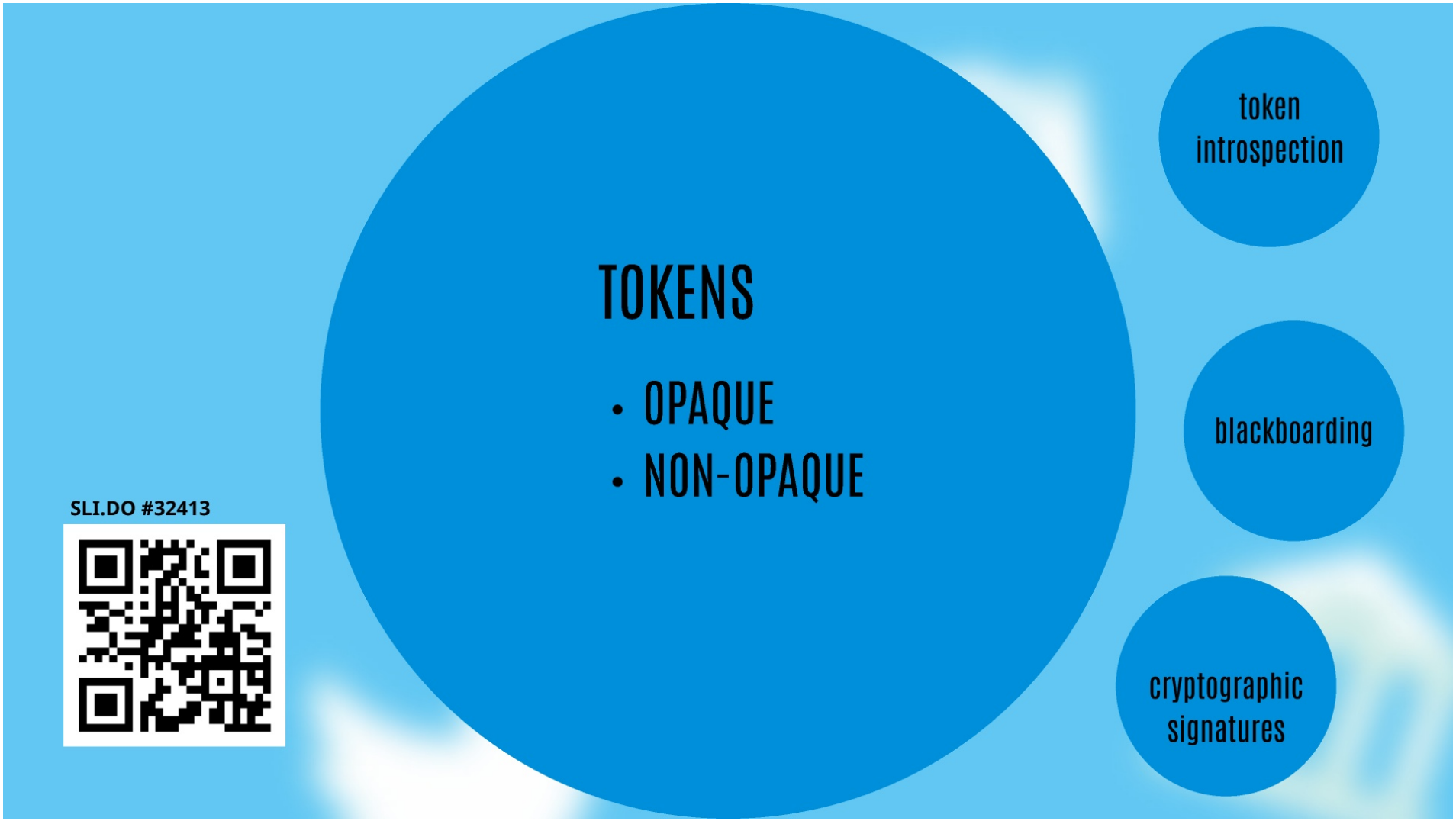

GRANT TYPES
• implicit grant type (deprecated)
• authorization code grant type
• password grant type (deprecated)
• client credentials grant type
AC
password
client
SLI.DO #32413

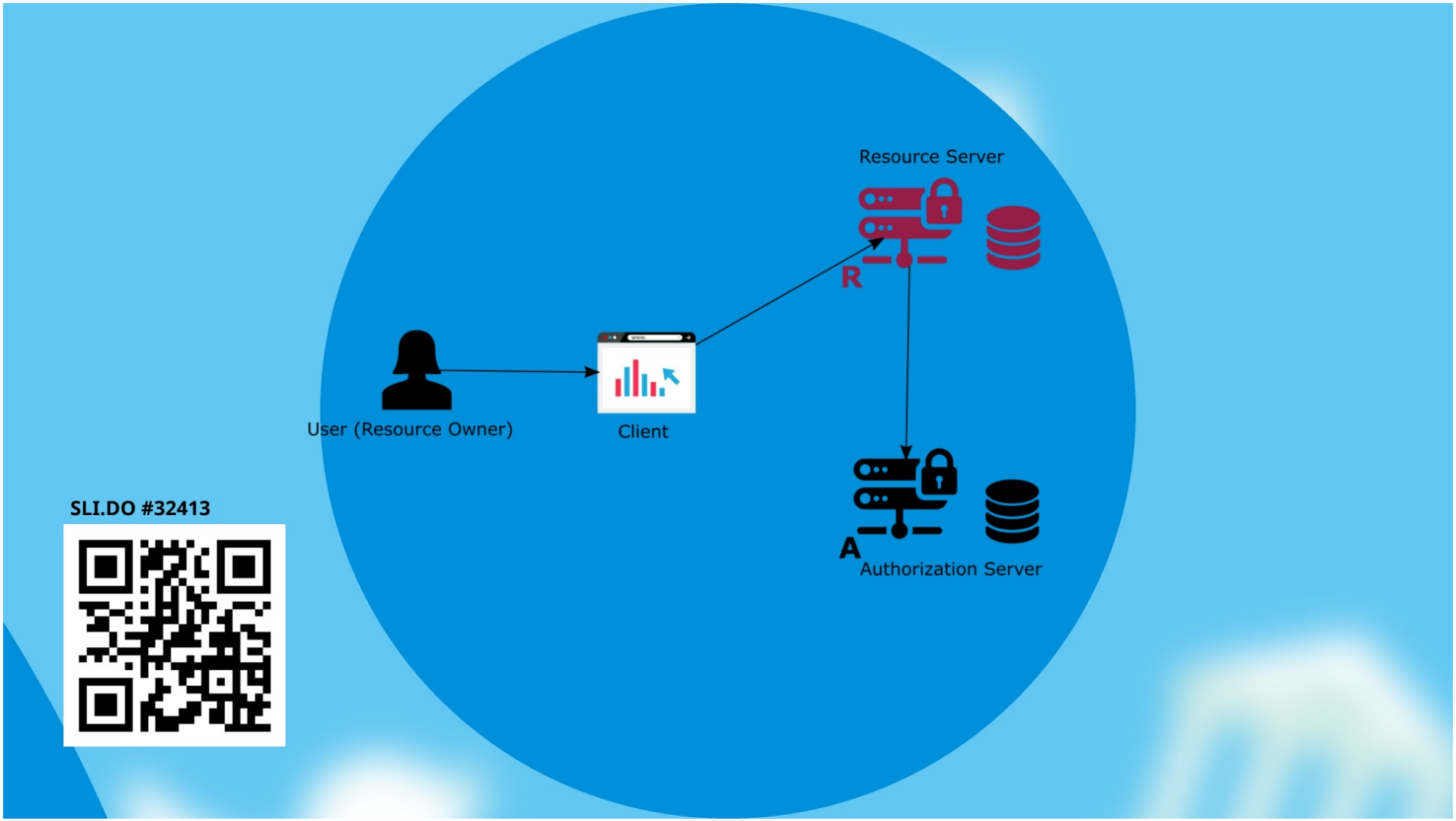

Client
Authorization Server
Resource Server
A
R
I need to access resource X.
I am not doing this on behalf of a user.
These are my client credentials.
Ok. I know your credentials.
Here is a token to access resource X.
I need to access resource X.
Here is a token from
the Authorization Server.
Your token is valid.
Here is resource X.

SLI.DO #32413

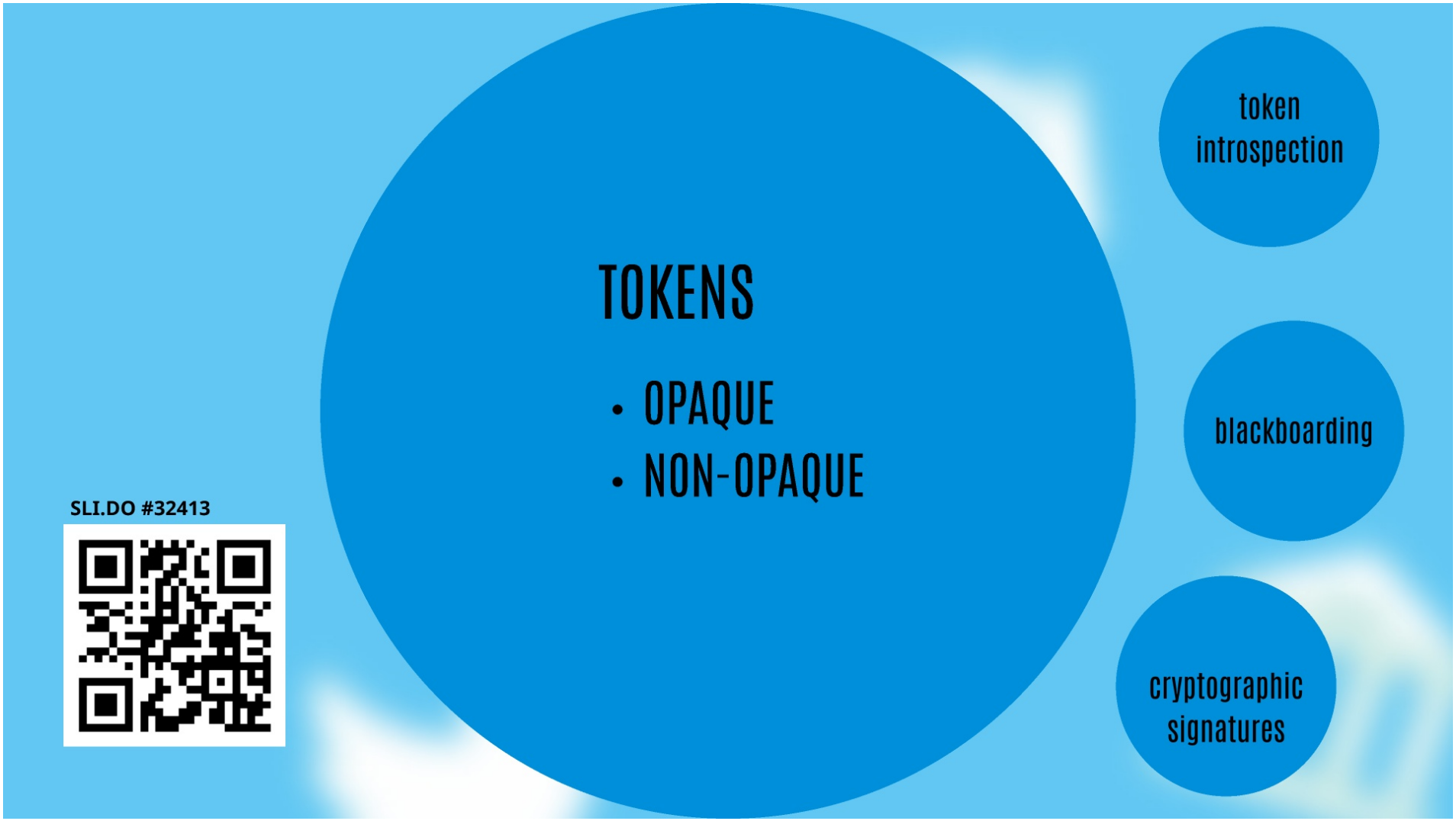

GRANT TYPES
• implicit grant type (deprecated)
• authorization code grant type
• password grant type (deprecated)
• client credentials grant type
AC
password
client
SLI.DO #32413

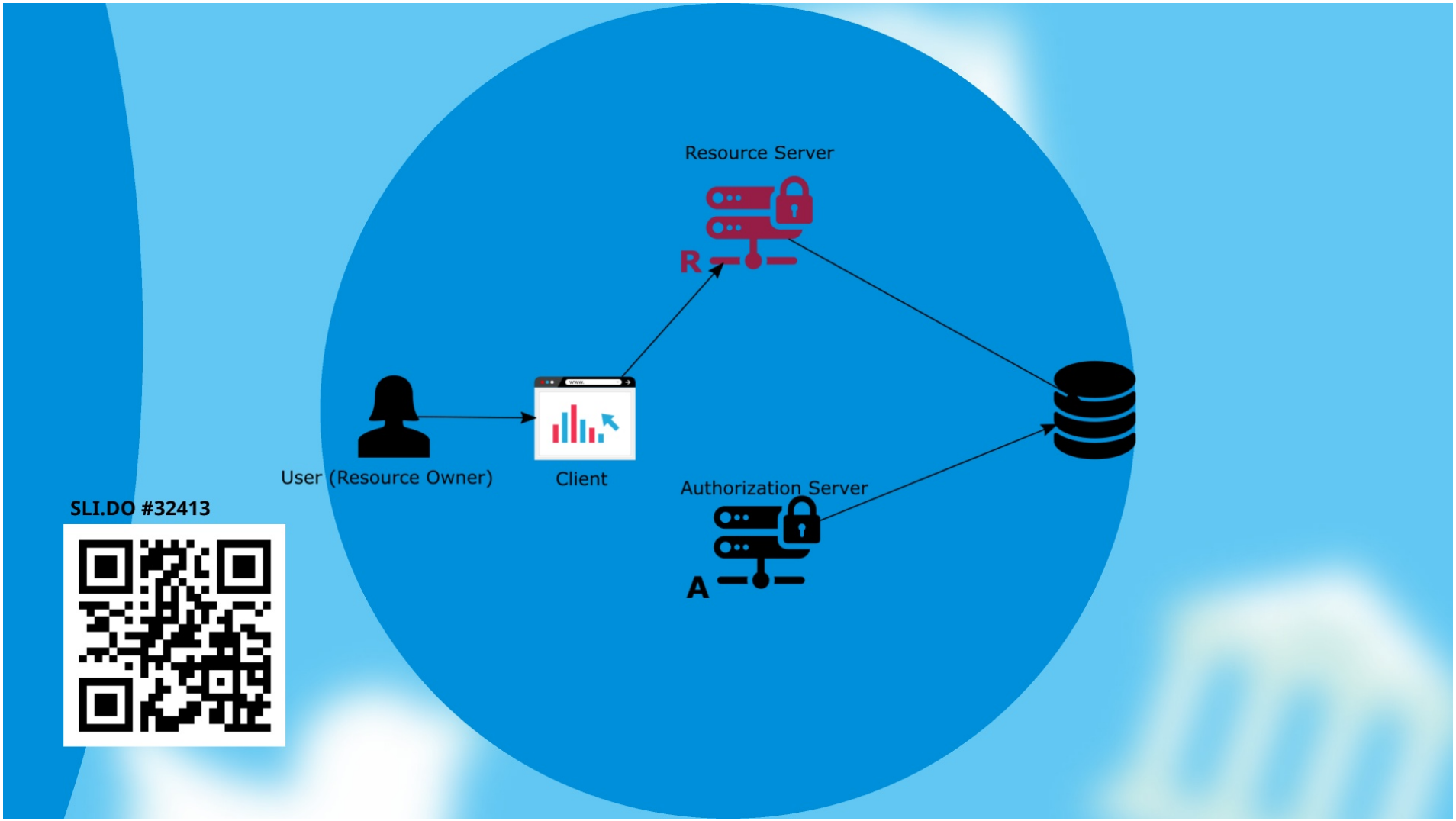

OAuth 2
What OAuth 2 is
• A specification
• A framework
What OAuth 2 isn't
• An implementation
SLI.DO #32413
The Actors
Flows
Tokens

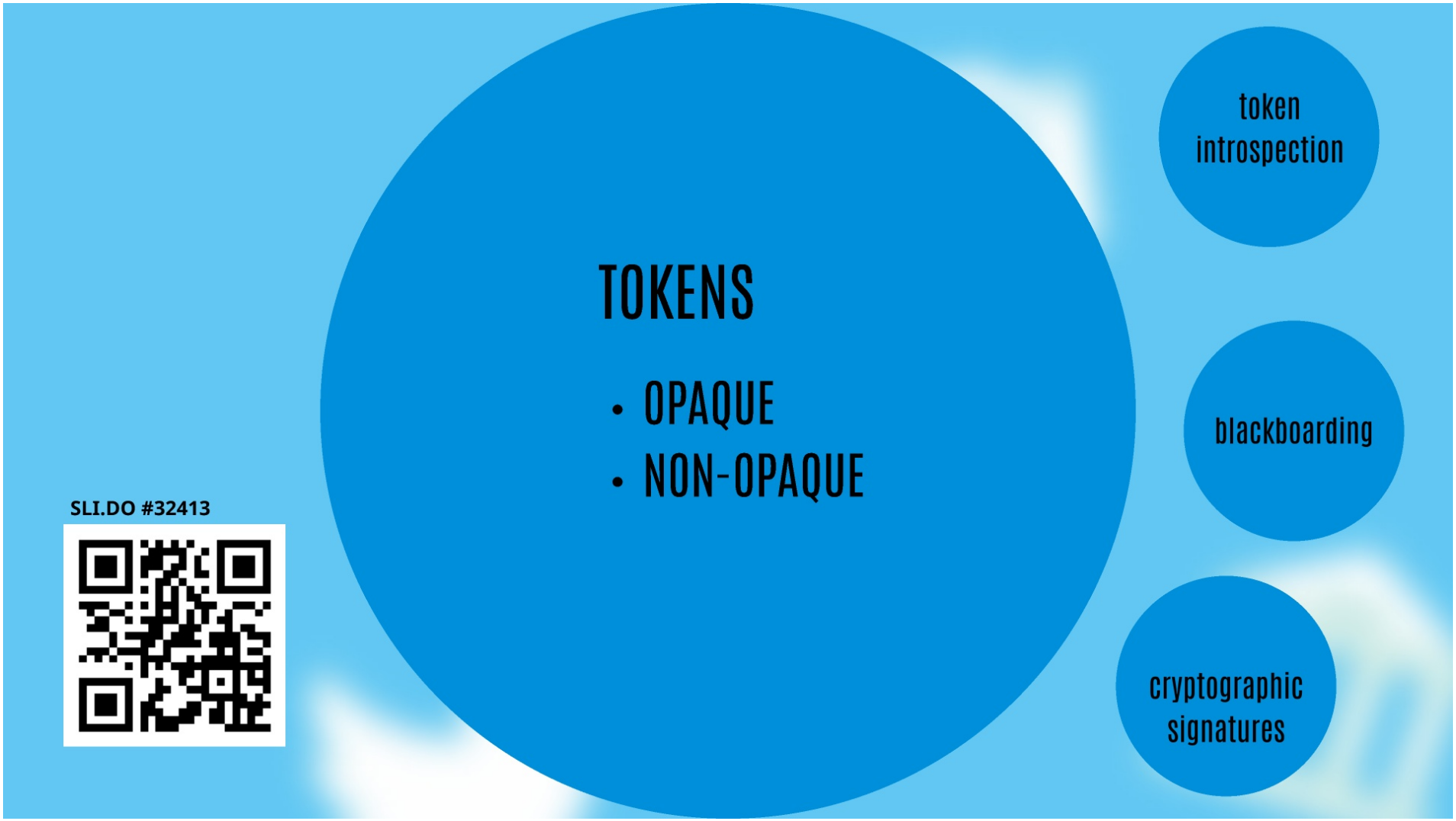

TOKENS
• OPAQUE
• NON-OPAQUE
token introspection
blackboarding
cryptographic signatures
SLI.DO #32413

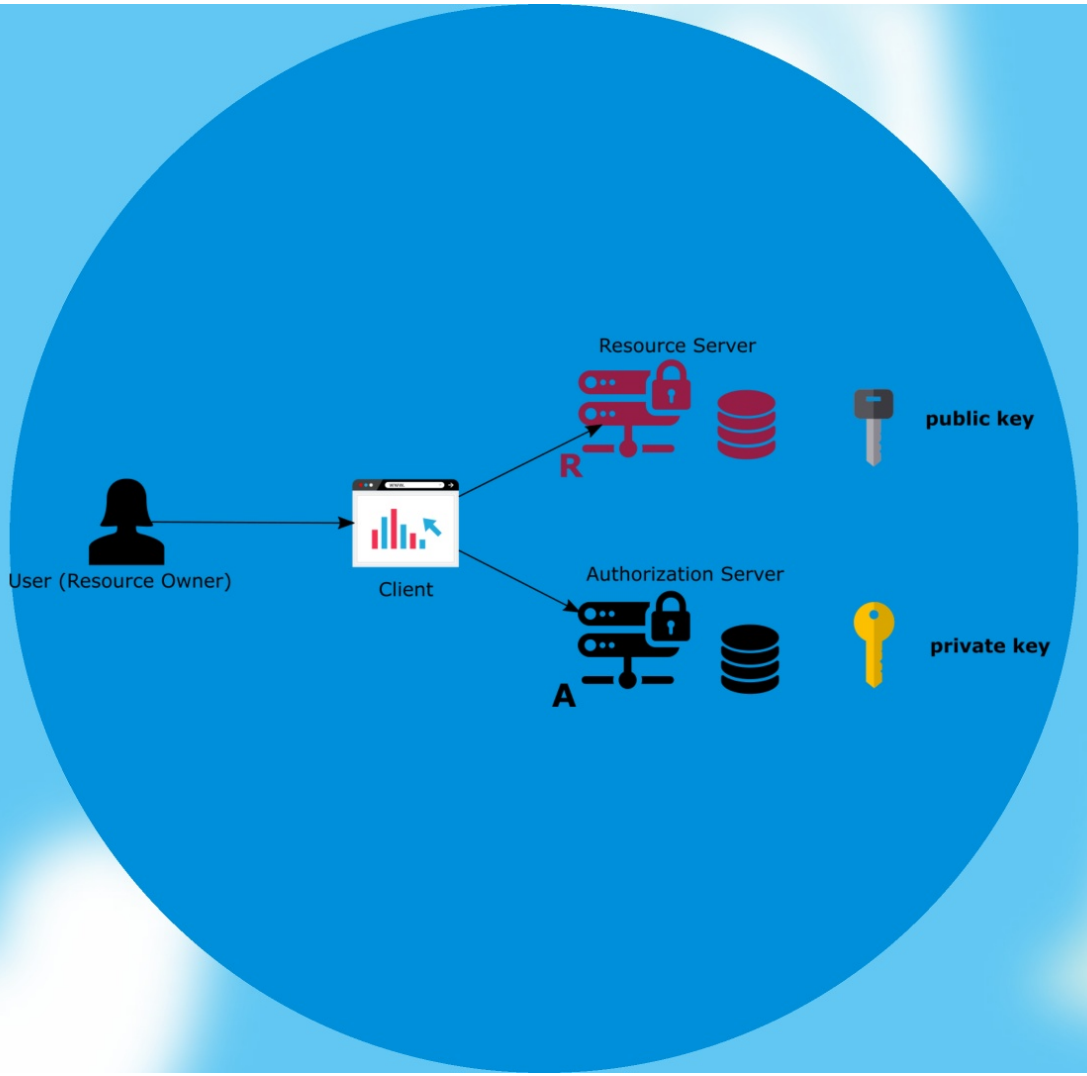

Resource Server
R
User (Resource Owner)
Client
A
Authorization Server

SLI.DO #32413

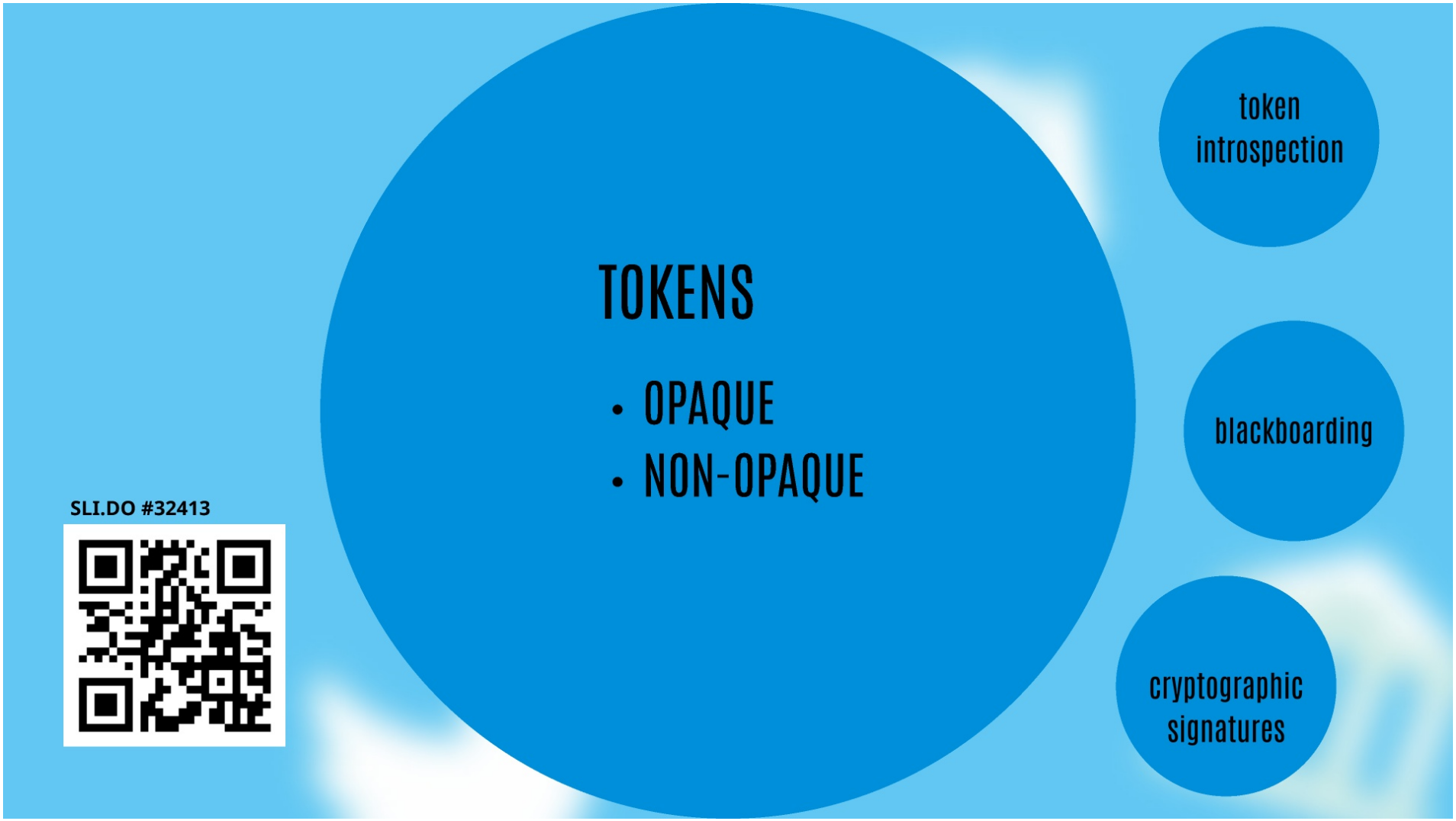

TOKENS
• OPAQUE
• NON-OPAQUE
token introspection
blackboarding
cryptographic signatures
SLI.DO #32413

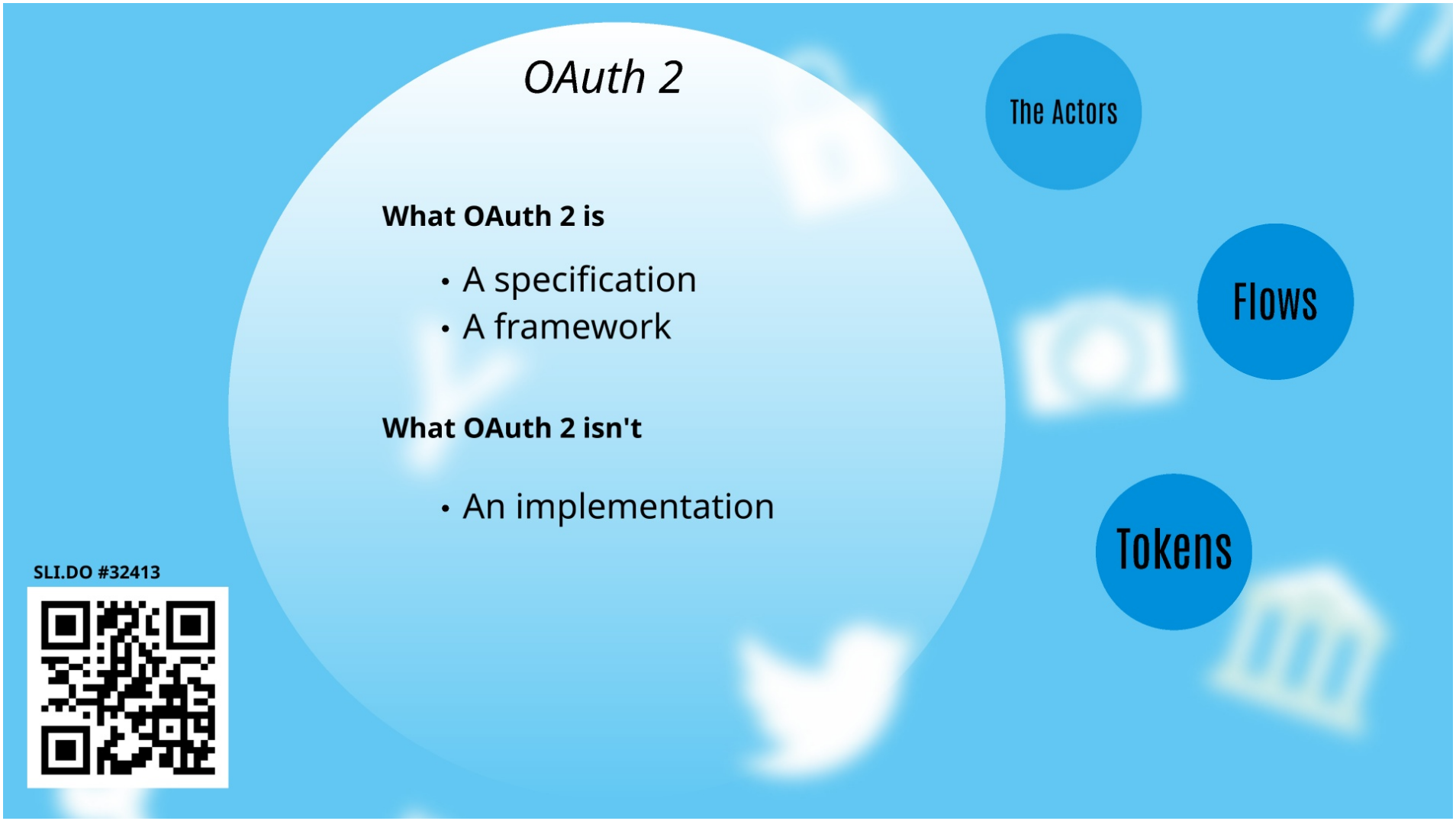

Resource Server
R
User (Resource Owner)
Client
Authorization Server
A
SLI.DO #32413


TOKENS
• OPAQUE
• NON-OPAQUE
token introspection
blackboarding
cryptographic signatures
SLI.DO #32413


Resource Server
public key
R
User (Resource Owner)
Client
Authorization Server
private key
A

SLI.DO #32413


TOKENS
• OPAQUE
• NON-OPAQUE
token introspection
blackboarding
cryptographic signatures
SLI.DO #32413


OAuth 2
What OAuth 2 is
• A specification
• A framework
What OAuth 2 isn't
• An implementation
SLI.DO #32413
The Actors
Flows
Tokens


endava
Security
How
Hands-on
OAuth 2
Using OAuth 2
With Spring Security
A Masterclass
SLI.DO #32413

Spring Security

- offers multiple ways to implement OAuth 2 systems
  - Spring Security OAuth
  - Spring Security

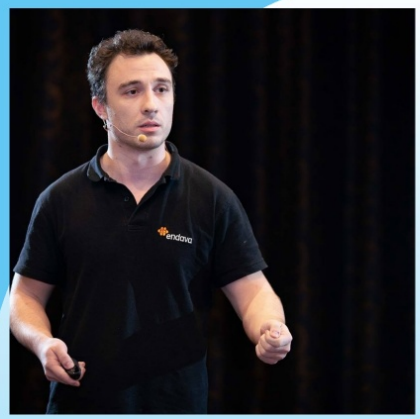
SLI.DO #32413

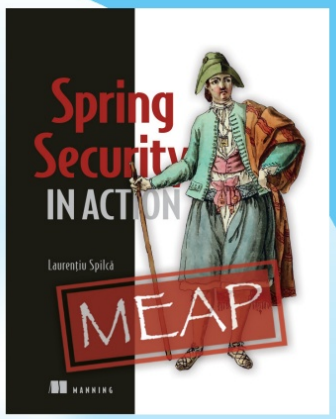

endava
How
Security
Hands-on
OAuth 2
Using OAuth 2
With Spring Security
A Masterclass
SLI.DO #32413

HANDS-ON

SLI.DO #32413


endava
How
Security
Hands-on
OAuth 2
Using OAuth 2
With Spring Security
A Masterclass
SLI.DO #32413





LAURENTIU SPILCA

Java Group Community Lead @ **Endava**

laurentiu.spilca@endava.com

SLI.DO #32413


endava
How
Security
Hands-on
OAuth 2
Using OAuth 2
With Spring Security
A Masterclass
SLI.DO #32413