Fast IDentity Online 2

What is it?

- It is a specification developed by W3C and the FIDO Alliance
- Specification of an authentication method that is much more secure and hassle free than passwords!

The components

A FIDO system has **3 components**

1. The Client
2. The Relying Party
3. The Authenticator

Authenticator

An authenticator

Client

A browser or an OS

Relying Party

The server

FIDO2 specification relies on communication between the three components

Authenticator

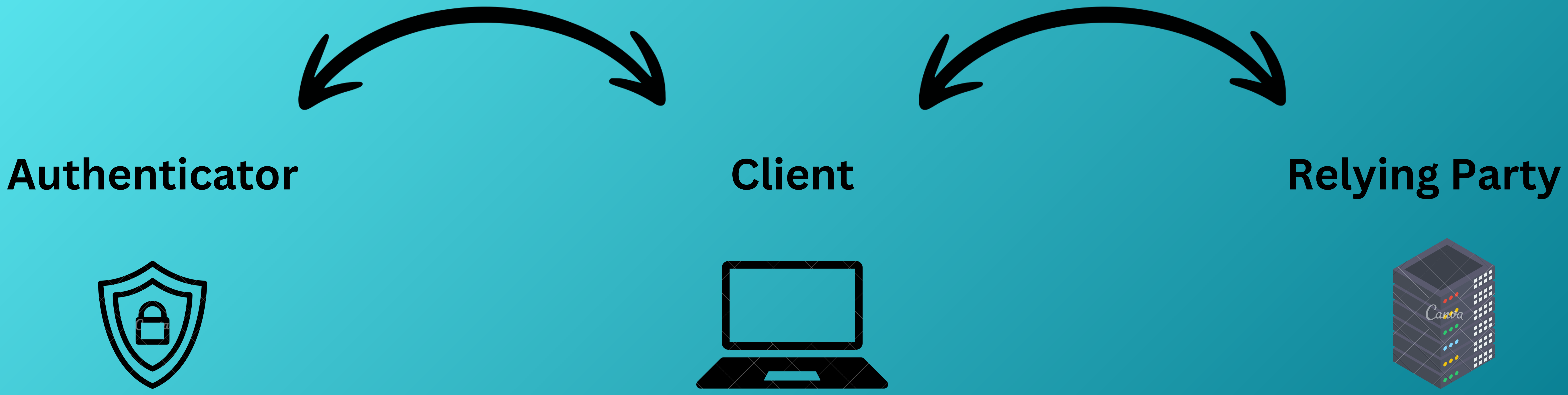
Client

Relying Party

The communication is modularised

One module for communication between Client and Authenticator

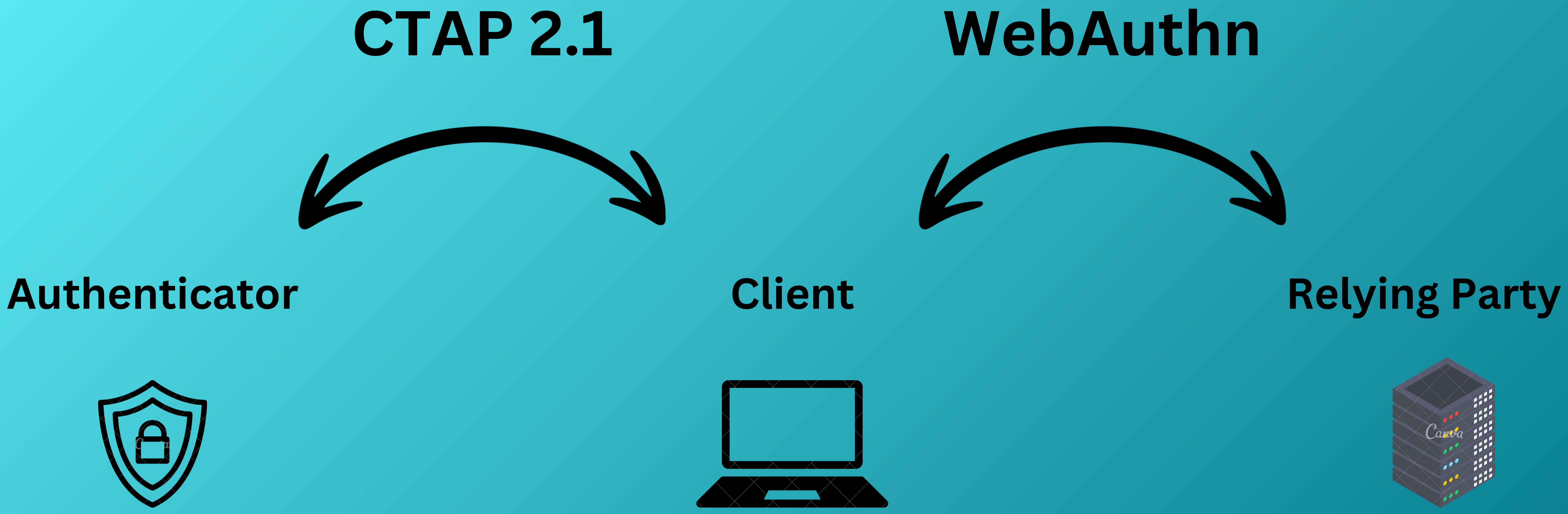
Another module for communication between Client and RP



Authenticator

Client

Relying Party



CTAP 2.1

WebAuthn

Authenticator

Client

Relying Party

Flows - Requests and Responses

The standard specifies flows for the two authentication processes

Registration Flow

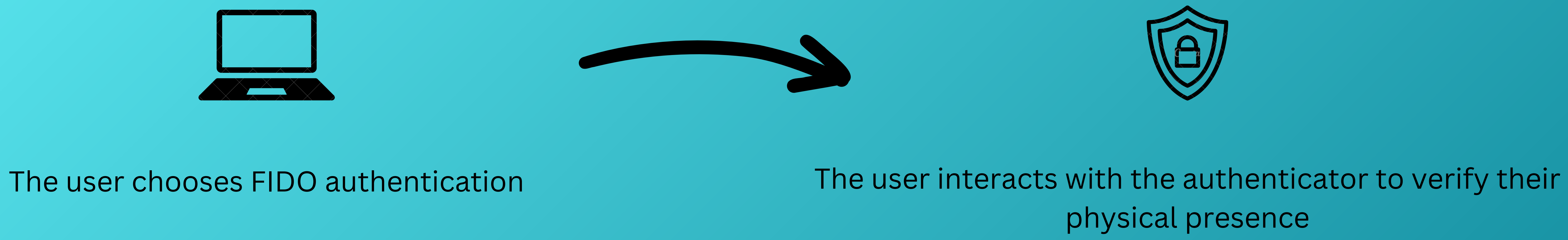
Login Flow

Registration Flow (On a high level)

Registration Flow (On a high level)

The user chooses FIDO authentication

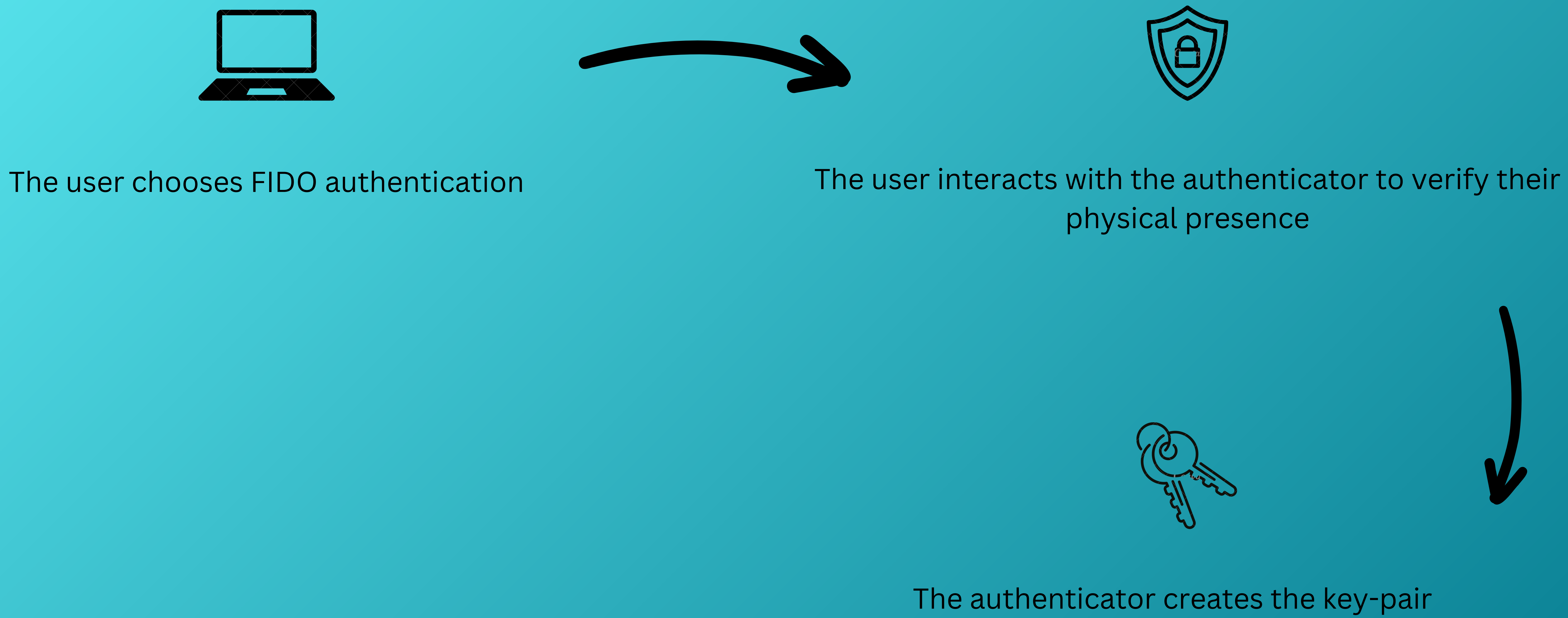
Registration Flow (On a high level)



The user chooses FIDO authentication

The user interacts with the authenticator to verify their physical presence

Registration Flow (On a high level)

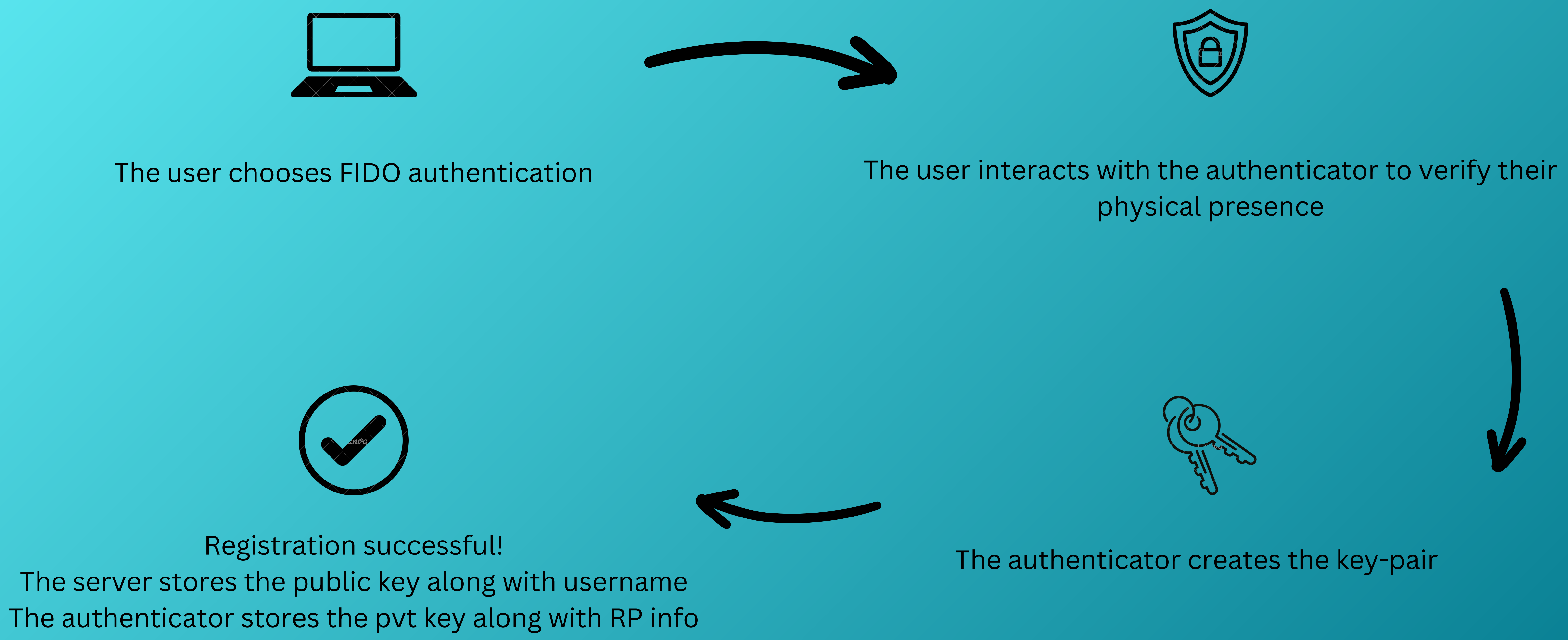


The user chooses FIDO authentication

The user interacts with the authenticator to verify their physical presence

The authenticator creates the key-pair

Registration Flow (On a high level)



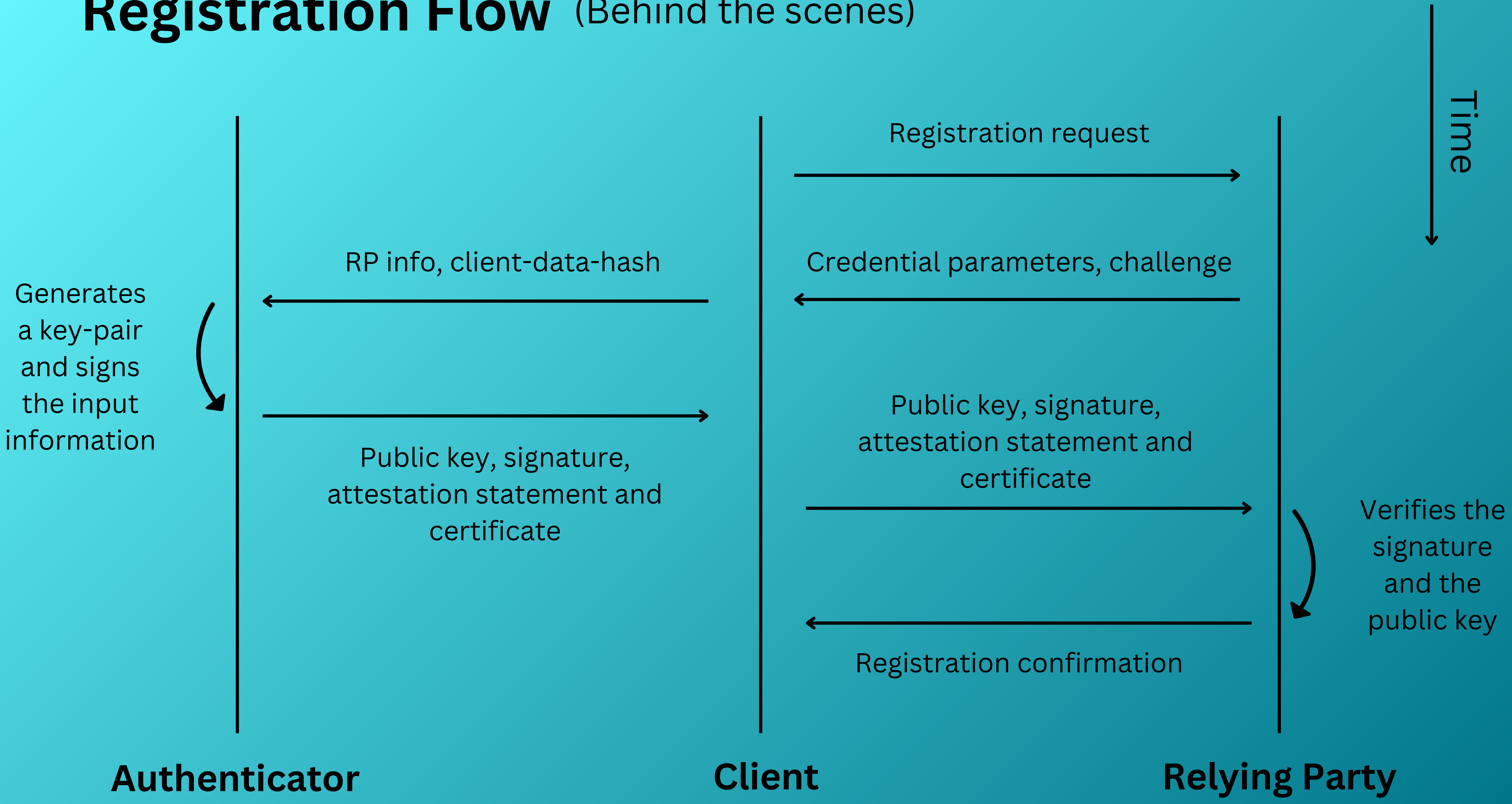
The user chooses FIDO authentication

The user interacts with the authenticator to verify their physical presence

The authenticator creates the key-pair

Registration successful!
The server stores the public key along with username
The authenticator stores the pvt key along with RP info

Registration Flow (Behind the scenes)

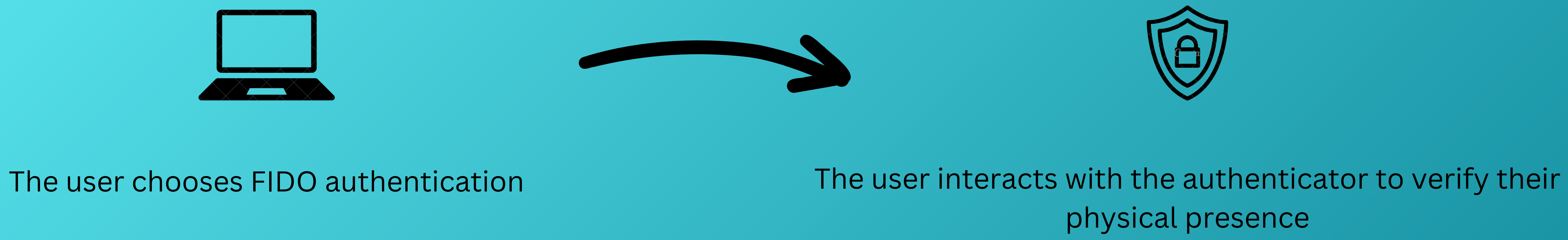

Time
Registration request
RP info, client-data-hash
Credential parameters, challenge
Generates a key-pair and signs the input information
Public key, signature, attestation statement and certificate
Public key, signature, attestation statement and certificate
Verifies the signature and the public key
Registration confirmation
Authenticator
Client
Relying Party

Login Flow (On a high level)

Login Flow (On a high level)

The user chooses FIDO authentication

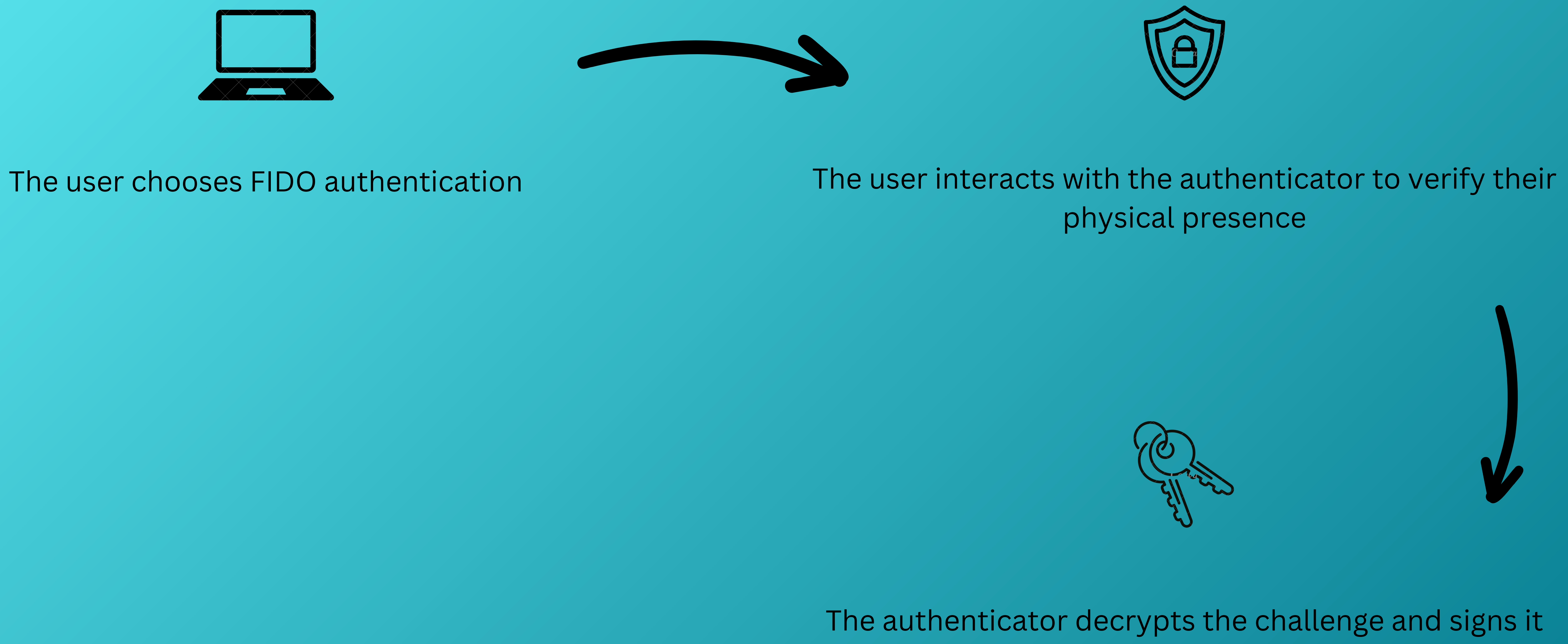
Login Flow (On a high level)



The user chooses FIDO authentication

The user interacts with the authenticator to verify their physical presence

Login Flow (On a high level)

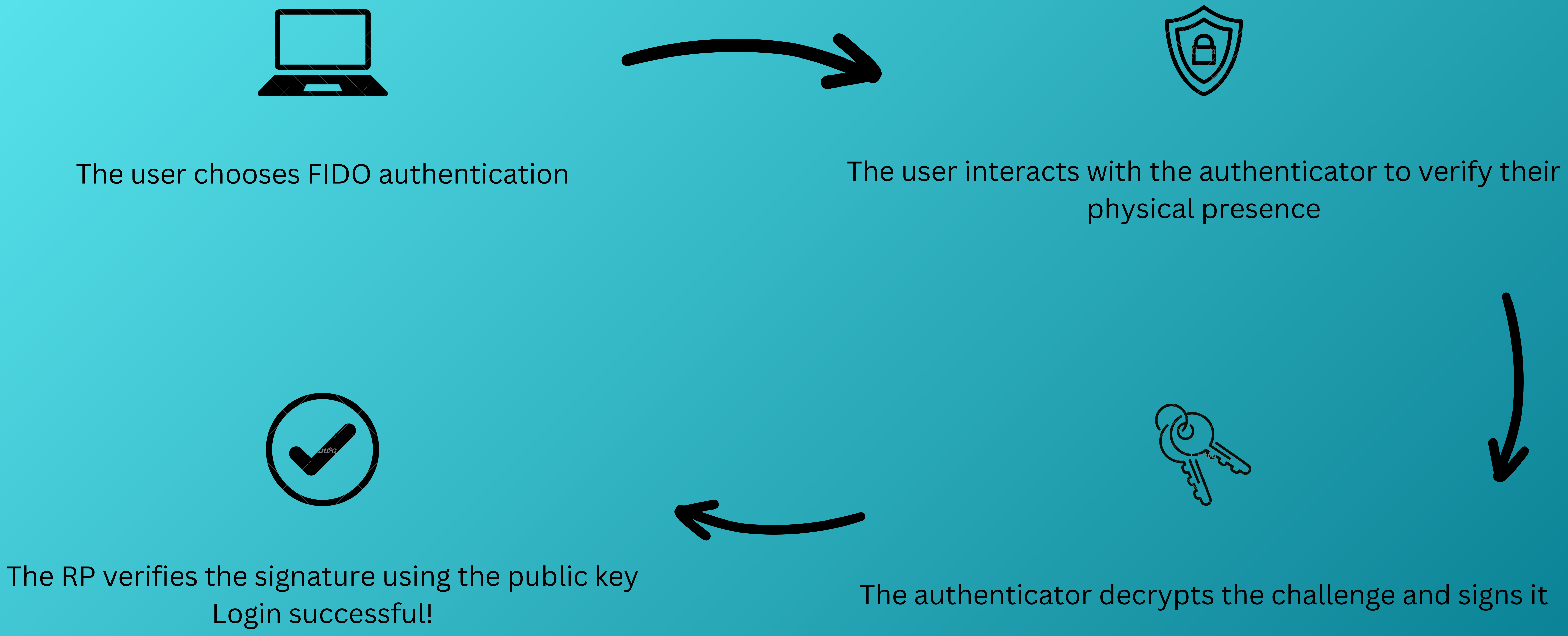


The user chooses FIDO authentication

The user interacts with the authenticator to verify their physical presence

The authenticator decrypts the challenge and signs it

Login Flow (On a high level)



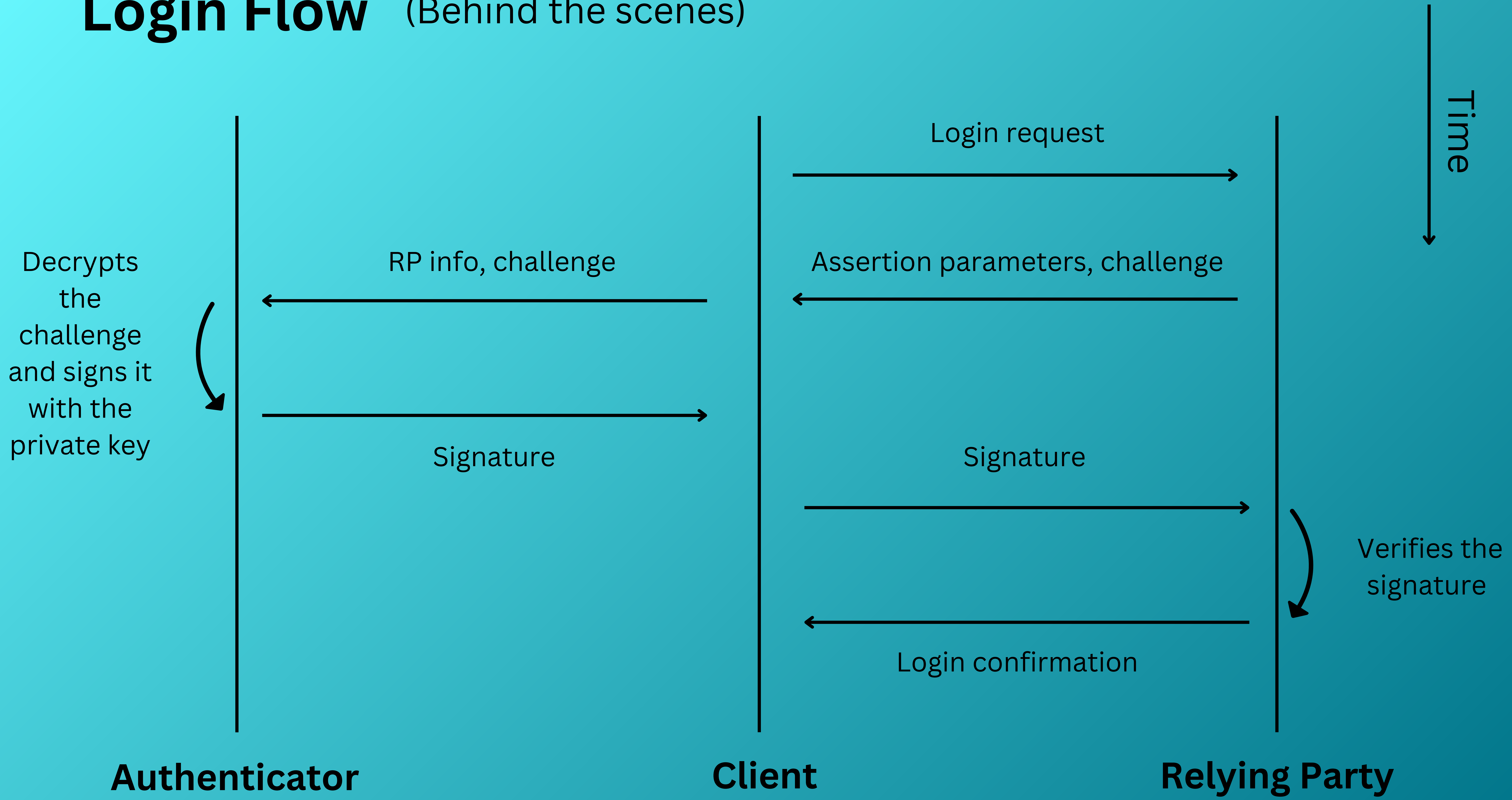
The user chooses FIDO authentication

The user interacts with the authenticator to verify their physical presence

The authenticator decrypts the challenge and signs it

The RP verifies the signature using the public key
Login successful!

Login Flow (Behind the scenes)



Time

Login request

RP info, challenge

Assertion parameters, challenge

Decrypts the challenge and signs it with the private key

Signature

Signature

Verifies the signature

Login confirmation

Authenticator

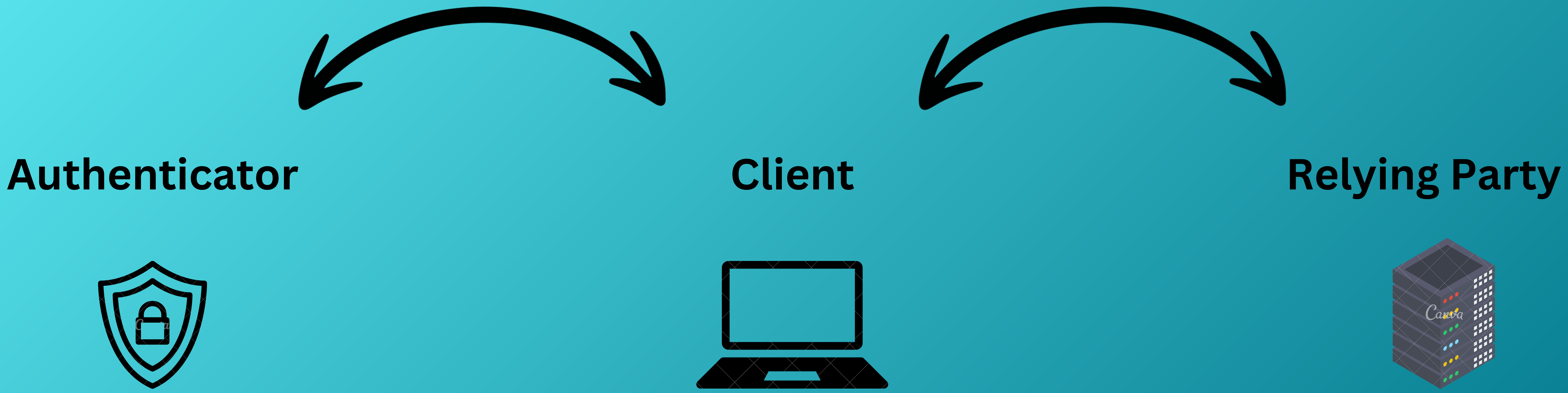
Client

Relying Party

The communication is modularised

One module for communication between Client and Authenticator

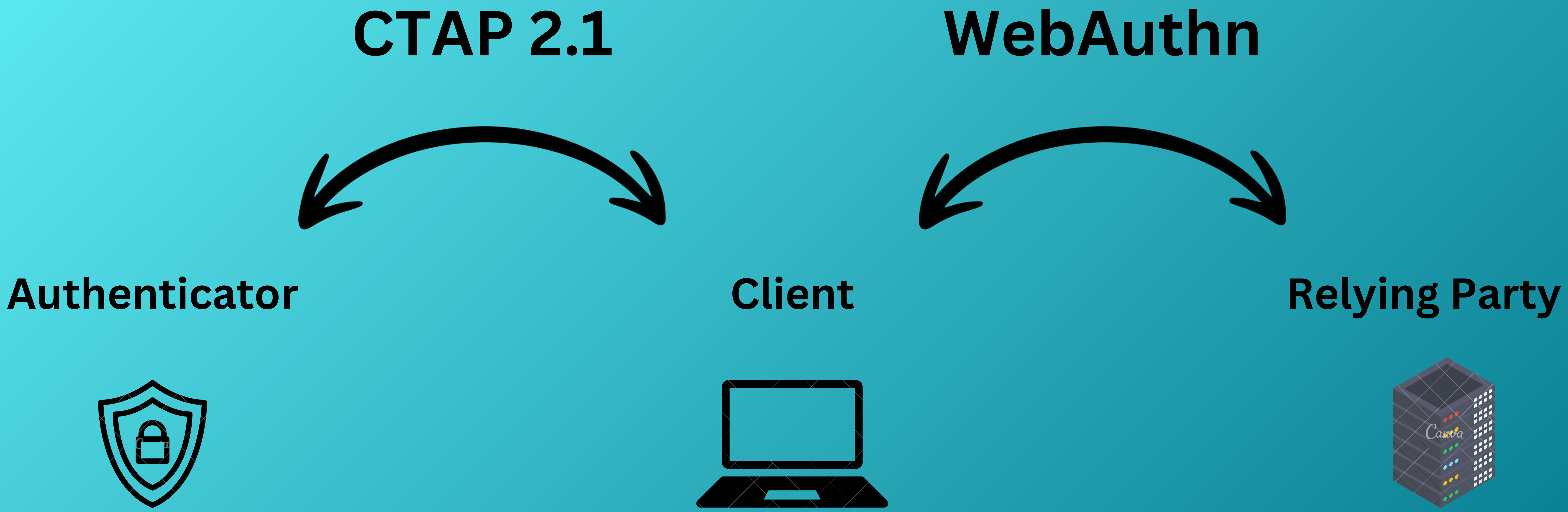
Another module for communication between Client and RP



Authenticator

Client

Relying Party



CTAP 2.1

WebAuthn

Authenticator

Client

Relying Party

WebAuthn

Web Authentication

Client

Relying Party

What is Web Authentication?

This is a broader term that refers to the process of verifying the identity of a user who is attempting to access a website. This is done by the website to protect their user's data from replay attacks.

This process ensures that the user is who they claim to be and it involves various mehods such as tockens, biometrics, multi-factor authentication.

What is WebAuthn?

This is refers to a specific standard developed by World Wide Web Consortium(W3C) and fido. It is a credential management API that is built in Web browsers.

This software allows users to register and authenticate with web applications using an authenticator such as a phone, hardware security keys in form usb sticks or bluetooth devices .

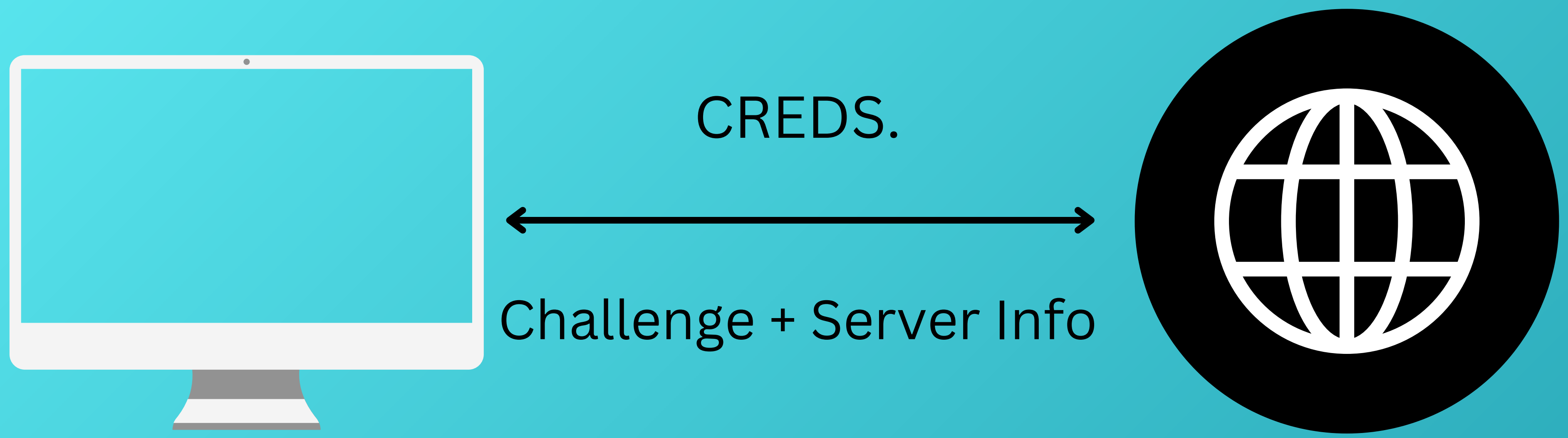
Recall the Stages When User Interacts With The application or service

1. Registration

2.Login

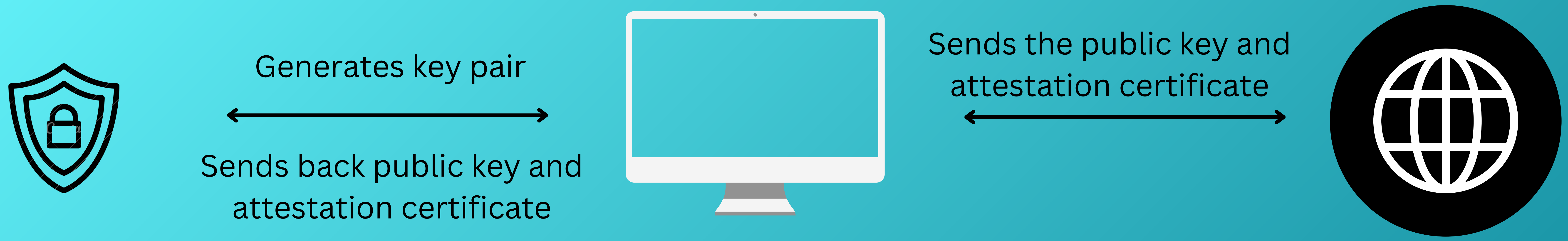
Where does WebAuthn fit in these cases?

1. **Registration:** The user signs in with the web server by entering the required credentials



The web server or RP then generates a unique challenge and sends back to the client along with Server id and other important parameters.

Authenticator

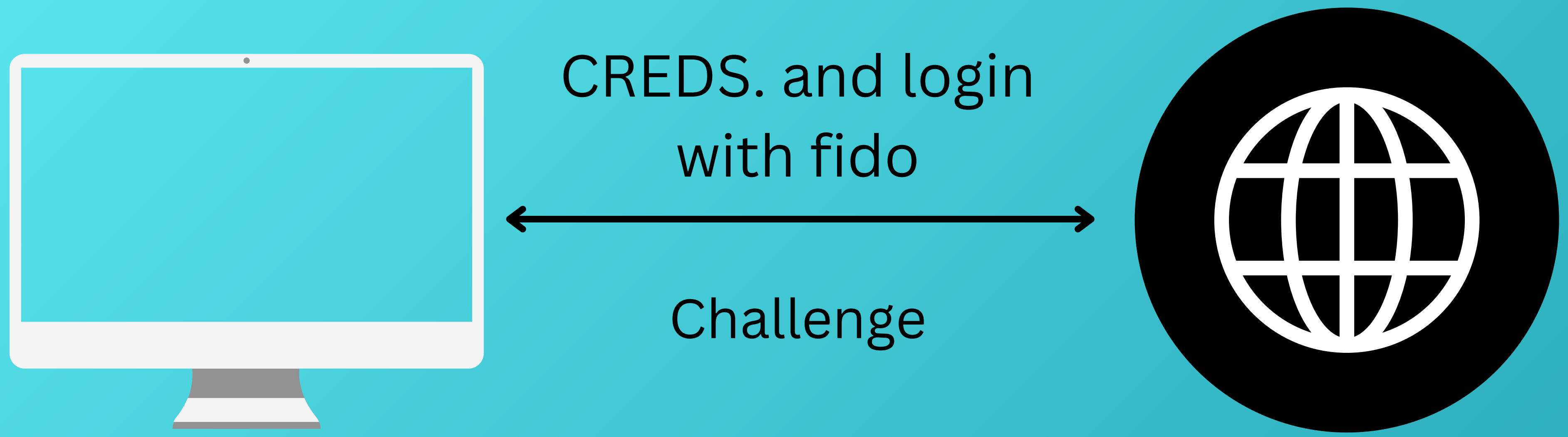

Generates key pair
Sends back public key and attestation certificate
Sends the public key and attestation certificate

Client then interacts with the authenticator, the authenticator generates key-pair. Client sends this registration data(public key and attestation certificate) to the relying party server.

If valid, the relying party stores the registration data and the public key

2. Login:

The user requests to log-in with the replying party by entering it’s credentials and choosing to log-in with fido.

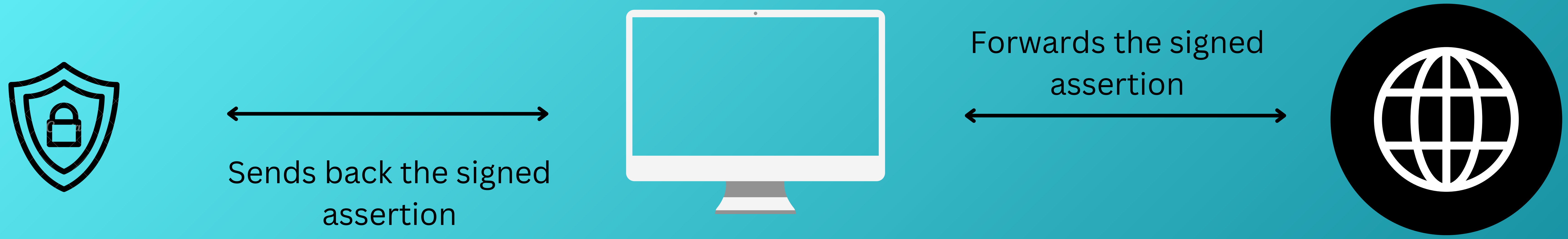


CREDS. and login with fido

Challenge

The web server or RP then generates a challenge and sends back to the client along with Server id .

The user selects a device for authentication.

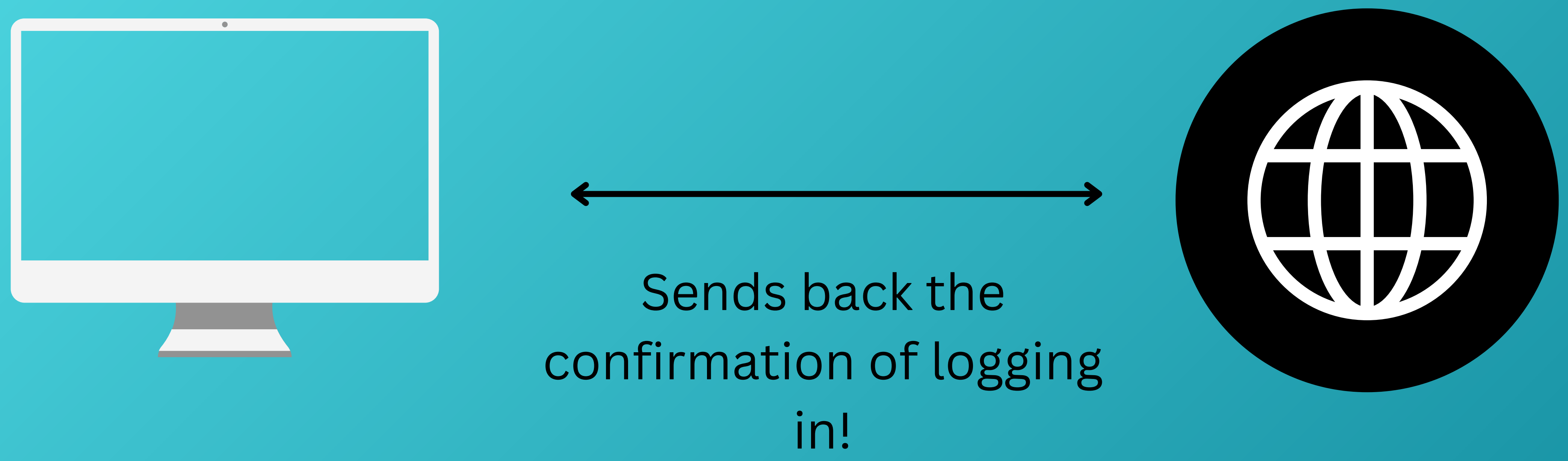

Sends back the signed assertion
Forwards the signed assertion

The selected authenticator uses it’s stored private key to sign the authentication data which involves authentication challenge generated by the server or relying party.

The fido authenticator creates a signed assertion using it’s private key, this is sent back to the client.

Final

The last stage involves the relying party server retrieving the stored public key



Sends back the confirmation of logging in!

Relying party server verifies the authenticity of signed assertion by using public key to decrypt the signature and validate that it matches the data stored with the RP server.

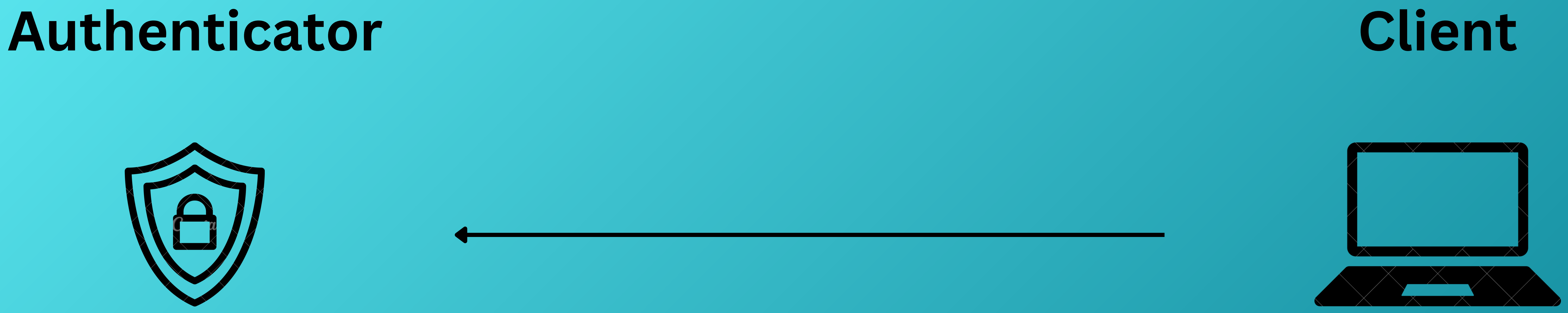
CTAP 2.1

Client To Authenticator Protocol v2.1

Authenticator

Client

Registration (Making a Credential)



Authenticator

Client

1. Credential parameters are sent to the authenticator

1. Credential parameters are sent to the authenticator

An obvious parameter is an RP identifier.

The parameters that are **necessary** as per the FIDO standard are -

1. Client Data Hash
2. Relying Party
3. User Entity
4. Algorithms

1. Credential parameters are sent to the authenticator

The following are the **optional** parameters -

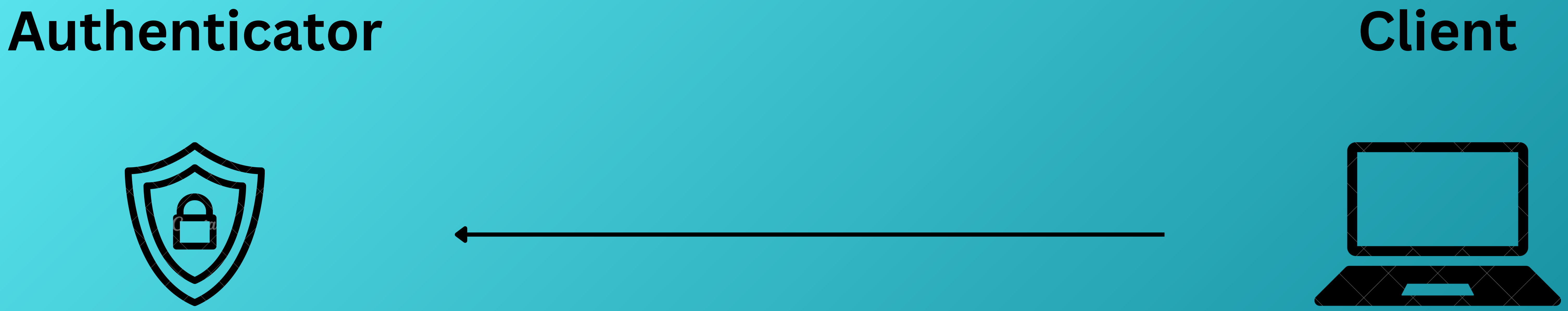
1. Protocol
2. PinUvAuthParam

Related to authenticator PIN

3. Exclude List
4. Extensions
5. Options
6. Enterprise Attestation

(Some models necessitate having a PIN factor for activating the authenticator. In such models, these parameters are necessary. eg - YubiKey)

Registration (Making a Credential)

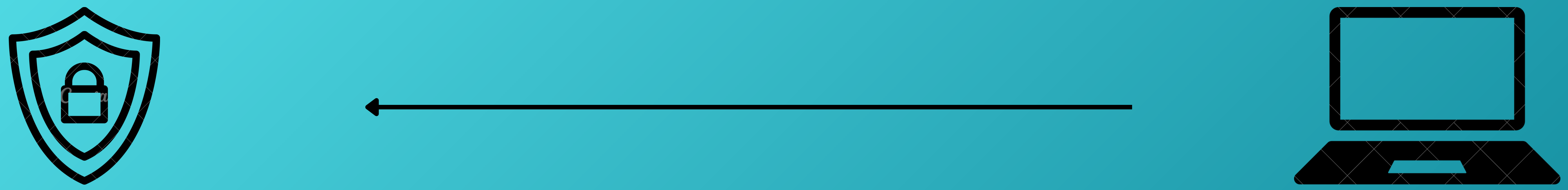


Authenticator

Client

2. The authenticator generates a key-pair, signs the input information and sends the public key along with attestation statement and certificate to the Client ,which is then relayed to the RP

Login (Getting an assertion)



Authenticator

Client

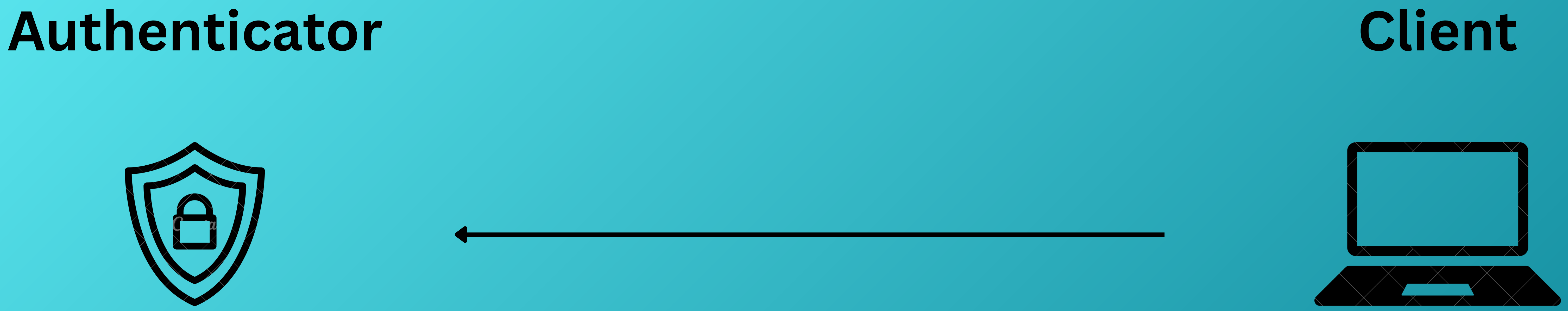
1. Assertion parameters are sent to the authenticator.

1. Assertion parameters are sent to the authenticator

The parameters that are **necessary** as per the FIDO standard are -

1. Client Data Hash
2. Relying Party

Login (Getting an assertion)



Authenticator

Client

2. The authenticator then looks up its storage for the private key associated with the Relying Party. This key is used to sign the Client Data Hash.

This signature is verified by the RP finishing the authentication.