certME

Digital identity using blockchain

www.certme.ro

certSIGN®

Presentation scope

Service description and ecosystem

Technology and data flows

User experience

Future work – IDBC project

What is certME?

And how it works?

Reusable ID verification for better UX and fresh data

registration and **authentication** to online services

transaction **authorization** or document **signing**

updating customer's **personal data**

Certified as an electronic means of identification

with a **substantial** level of assurance,

in accordance with **Regulation (EU) 2014/910**

and ADR **Decision 564/2021**

certME ecosystem

Roles and implications

Roles within the certME ecosystem

Scheme administrator – certSIGN company

Validator – Partner organization offering identity verification

Online service provider – Client organization using certME to authenticate and enroll users.

User – Natural person that agreed to certME T&C

How it works

blockchain under the hood

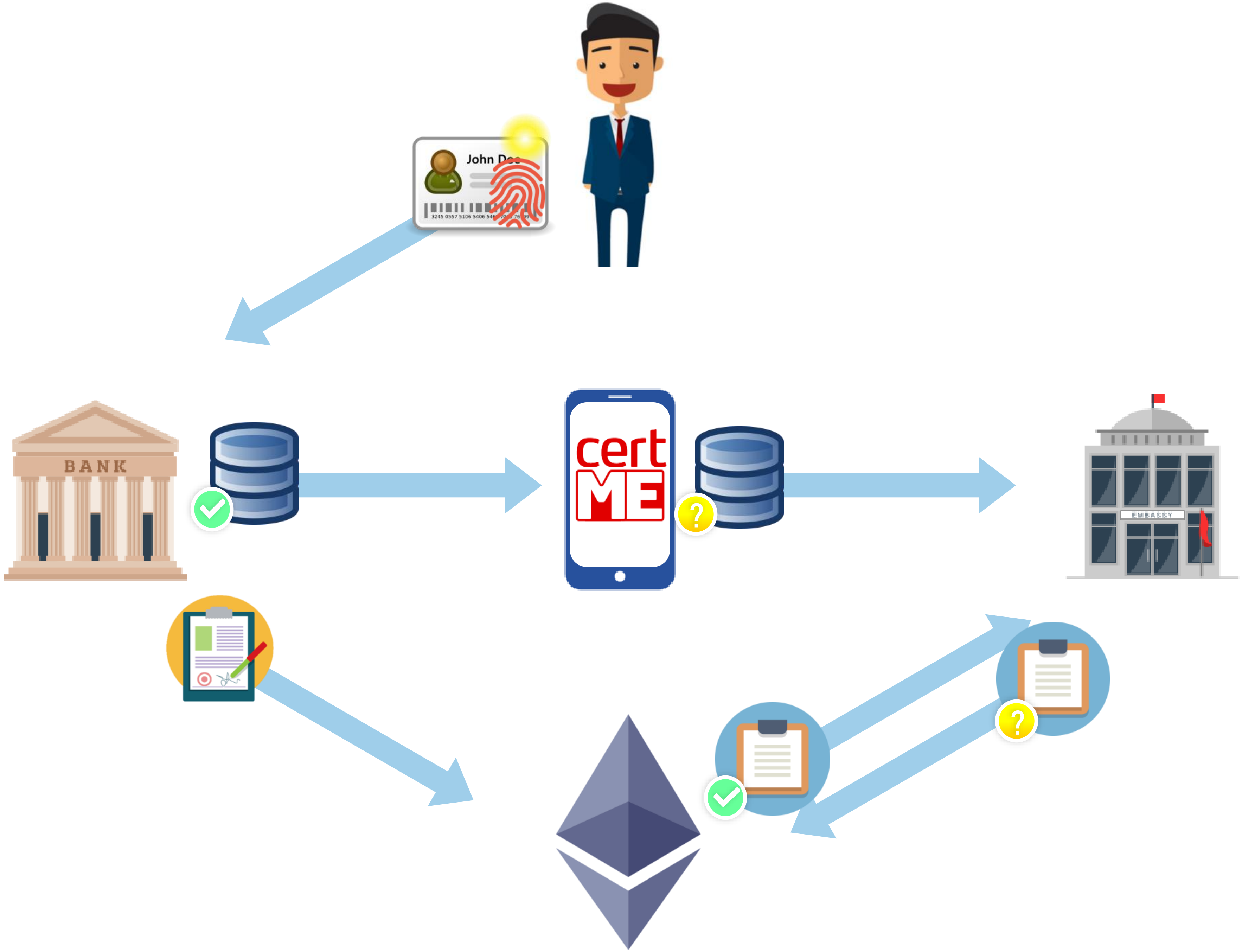
1. The user’s identity is verified by a certME validator

2. The validator stores the user’s data on the user’s device and stores proofs of verification on the blockchain

3. The user registers to a service provider by sending their data from the certME app

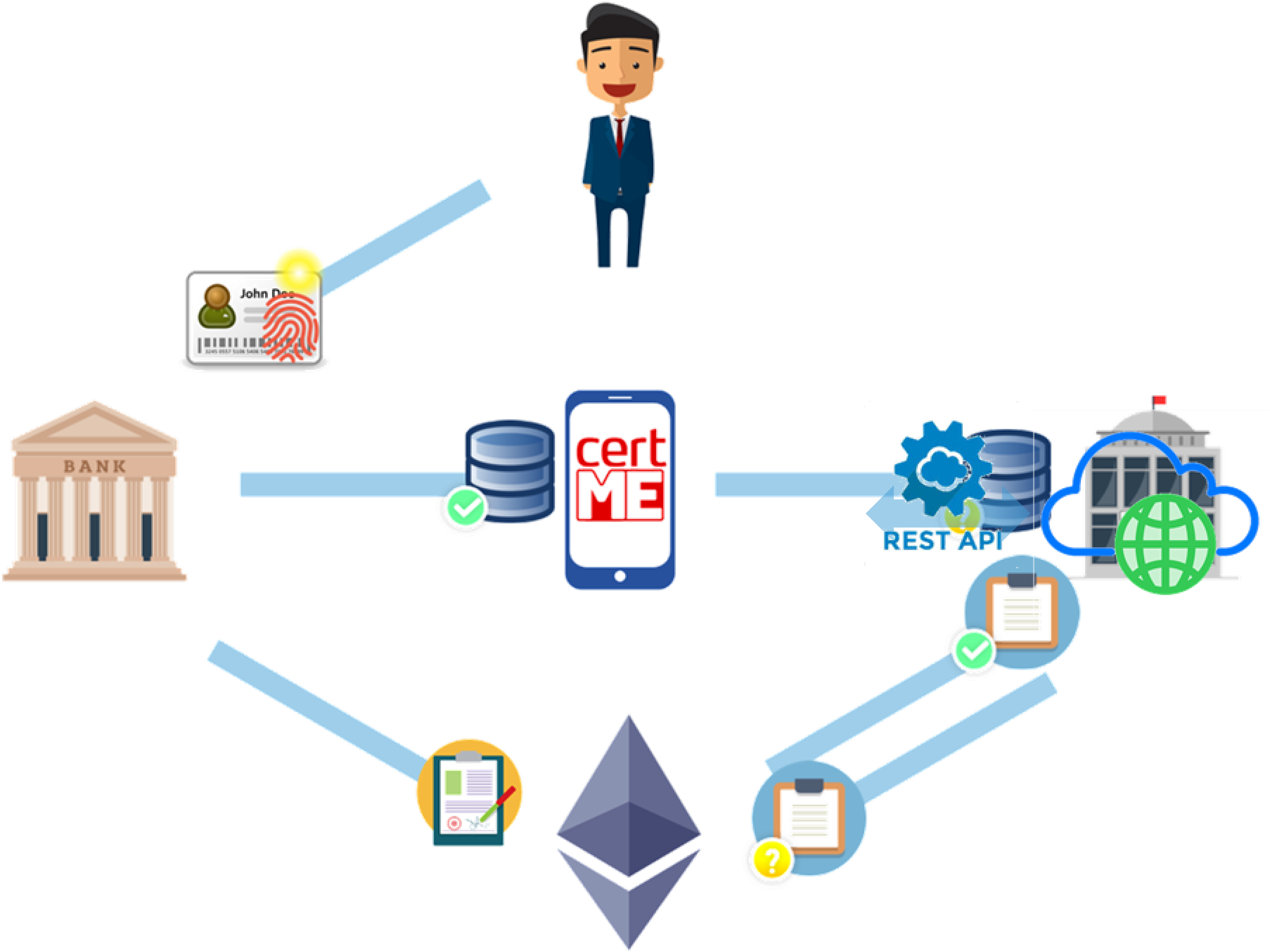
4. The service provider submits a validation request for the user’s data on the blockchain

5. The service provider receives a validation confirmation from the blockchain and registers the user

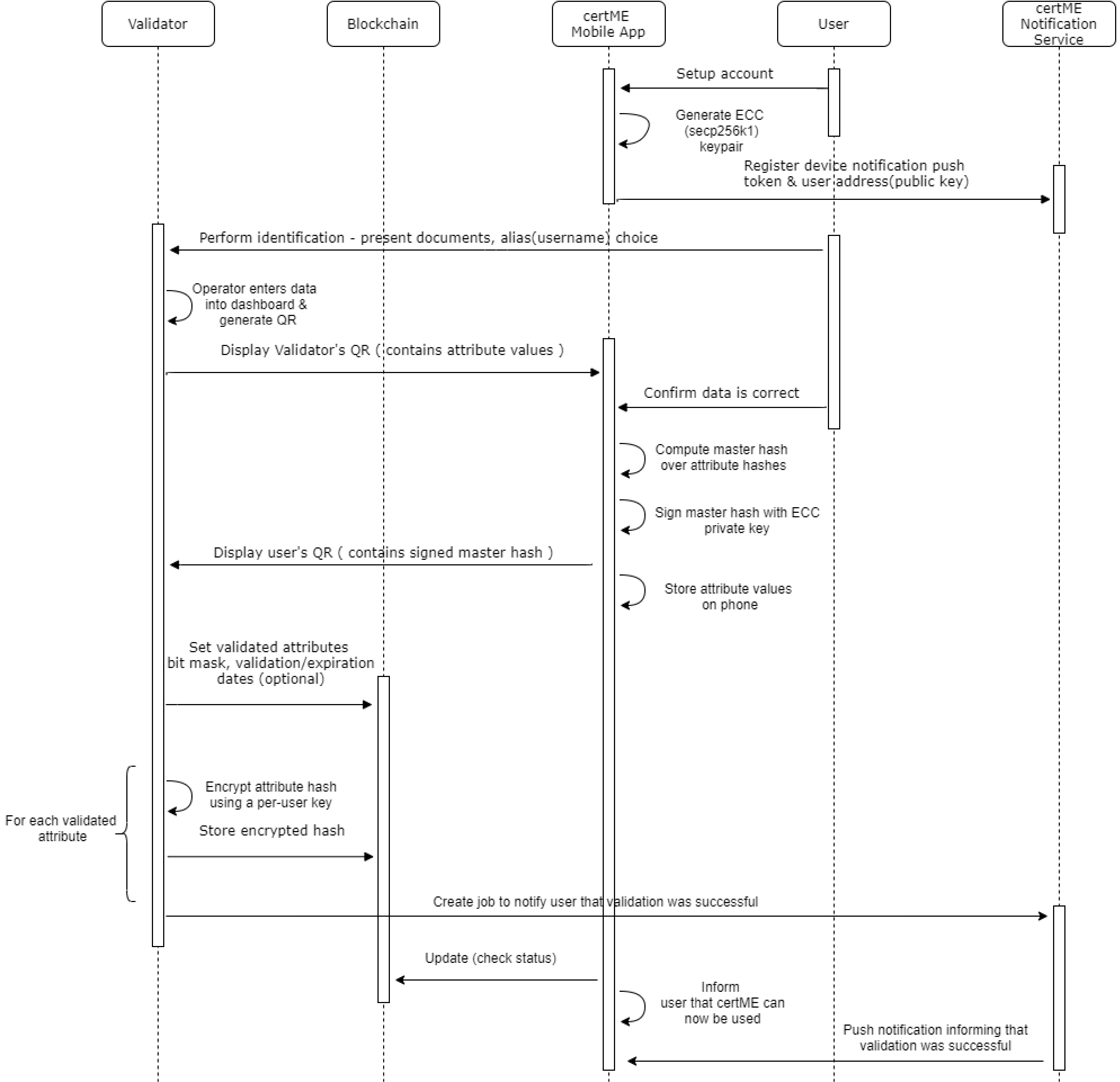


certME

1. The user's identity is verified by a certME validator
2. The validator stores the user's data on the user's device and stores proofs of verification on the blockchain
3. The user registers to a service provider by sending their data from the certME app
4. The service provider submits a validation request for the user's data on the blockchain
5. The service provider receives a validation confirmation from the blockchain and registers the user



certME



Validator

Blockchain

certME Mobile App

User

certME Notification Service

Setup account

Generate ECC (secp256k1) keypair

Register device notification push token & user address(public key)

Perform identification - present documents, alias(username) choice

Operator enters data into dashboard & generate QR

Display Validator's QR (contains attribute values)

Confirm data is correct

Compute master hash over attribute hashes

Sign master hash with ECC private key

Display user's QR (contains signed master hash)

Store attribute values on phone

Set validated attributes bit mask, validation/expiration dates (optional)

For each validated attribute

Encrypt attribute hash using a per-user key

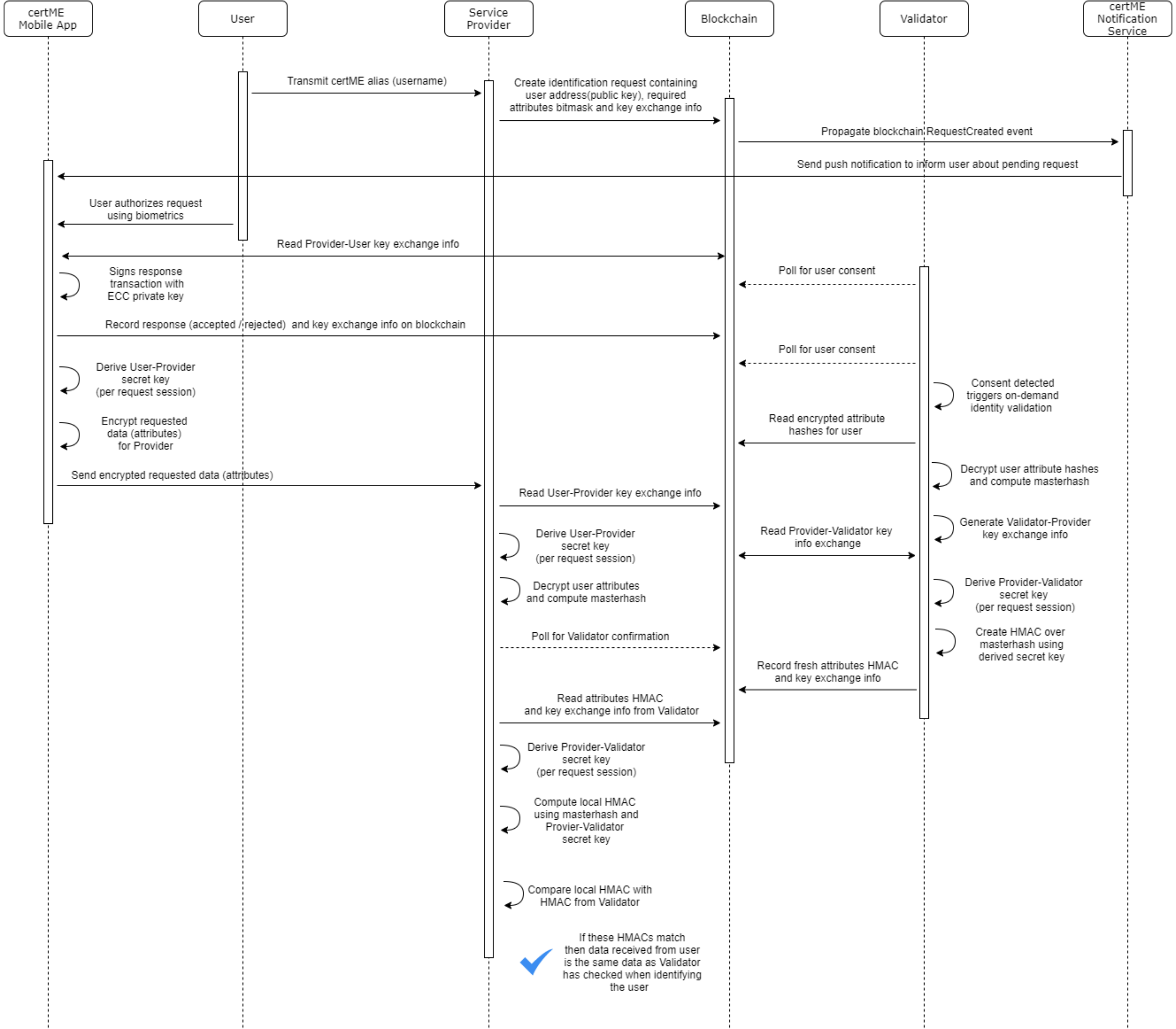
Store encrypted hash

Create job to notify user that validation was successful

Update (check status)

Inform user that certME can now be used

Push notification informing that validation was successful



certME Mobile App

User

Service Provider

Blockchain

Validator

certME Notification Service

Transmit certME alias (username)

Create identification request containing user address(public key), required attributes bitmask and key exchange info

Propagate blockchain RequestCreated event

Send push notification to inform user about pending request

User authorizes request using biometrics

Read Provider-User key exchange info

Signs response transaction with ECC private key

Poll for user consent

Record response (accepted / rejected) and key exchange info on blockchain

Poll for user consent

Derive User-Provider secret key (per request session)

Consent detected triggers on-demand identity validation

Encrypt requested data (attributes) for Provider

Read encrypted attribute hashes for user

Decrypt user attribute hashes and compute masterhash

Send encrypted requested data (attributes)

Read User-Provider key exchange info

Generate Validator-Provider key exchange info

Derive User-Provider secret key (per request session)

Read Provider-Validator key info exchange

Decrypt user attributes and compute masterhash

Derive Provider-Validator secret key (per request session)

Poll for Validator confirmation

Create HMAC over masterhash using derived secret key

Record fresh attributes HMAC and key exchange info

Read attributes HMAC and key exchange info from Validator

Derive Provider-Validator secret key (per request session)

Compute local HMAC using masterhash and Provier-Validator secret key

Compare local HMAC with HMAC from Validator

✓ If these HMACs match then data received from user is the same data as Validator has checked when identifying the user

certME

certME API

Interacting with certME via REST API

certME Provider Sign Requests API

Used for authentication and authorization

- Targetless mode
  - Used for authentication
  - Anyone can authenticate by scanning a QR or accessing a link
- Targeted mode
  - Used for authentication (e.g., recent logins) and authorization
  - Only the targeted user can respond to the request
- Authentication to API is done using mutual TLS with X509 client certificates

certME Provider Sign Requests API

Each service provider gets its own entry point.

- Step 1: Creating a sign request
  - POST <entry-point>/eids/:eID/sign-requests
  - The response includes a link that the user must access for authentication and a QR that includes the link
  - The response header includes a URL that will be used for polling to obtain the authentication status
- Step 2: Polling to obtain status -> repeats until expiration, or sign/cancel event
  - GET <URL returned in response header from creation>
  - The response contains a list of events (access – the code was scanned, sign – the user accepted the authentication, cancel – the user canceled the authentication)

certME Provider Data Requests API

Used to request user data (when enrolling or updating data)

- Attributes currently supported: Name, Surname, Date of birth, CNP, Place of birth, Domicile, Gender, Citizenship, Document type, Document issuer, Document series, Document number, Document issue date, Document expiration date, Country, Personal phone number
- Each certME digital identity is identified by
  - Id – unique identifier of the person in the certME system (similar to the CNP) which remains unchanged even if the certME digital identity is re-issued or updated
  - Address – unique identifier of the certME digital identity (similar to serial/CI no) that changes if the certME digital identity is re-issued or updated
- API authentication is done using mutual TLS with X509 client certificates

certME Provider Data Requests API

Each service provider gets its own entry point.

- Step 0 (optional): Authenticate the user to get the ID
- Step 1: Creating a data request
  - POST <entry-point>/eids/:eID/data-requests
  - The body includes the attributes requested from the user
  - The response header includes a URL that will be used for polling to obtain the status of the request
- Step 2: Polling to obtain status -> repeats until expiration, or date/reject event
  - GET <URL returned in response header from creation>
  - The response contains a list of events (accept, acceptConfirmed, data, reject, rejectConfirmed, error) and the personal data requested from the user (if the "data" event was received)

User experience

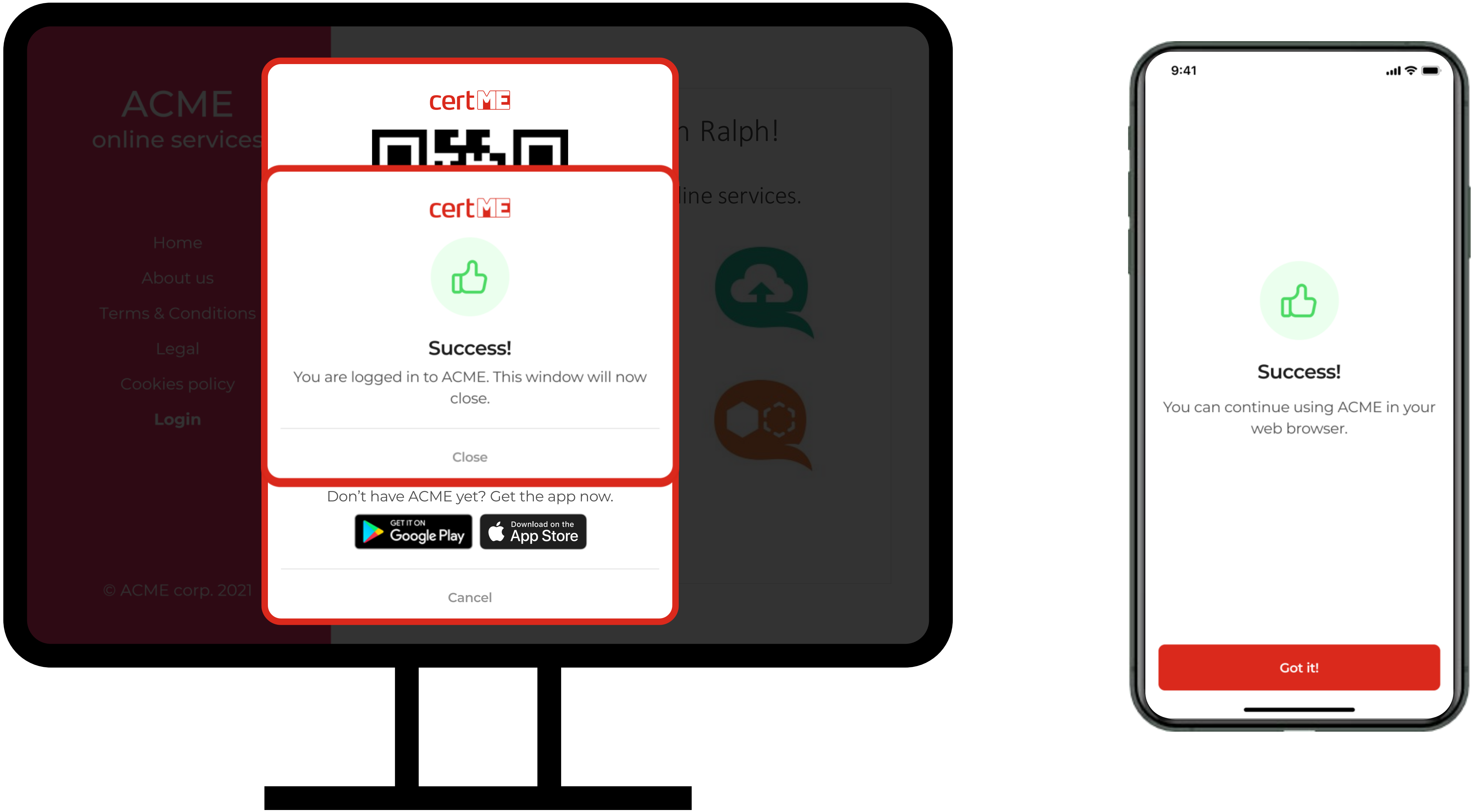
Authentication

Registration

On the fly eID issuance & registration

Existing customer authentication

registered customer login with certME



Success!

You are logged in to ACME. This window will now close.

Don't have ACME yet? Get the app now.

Success!

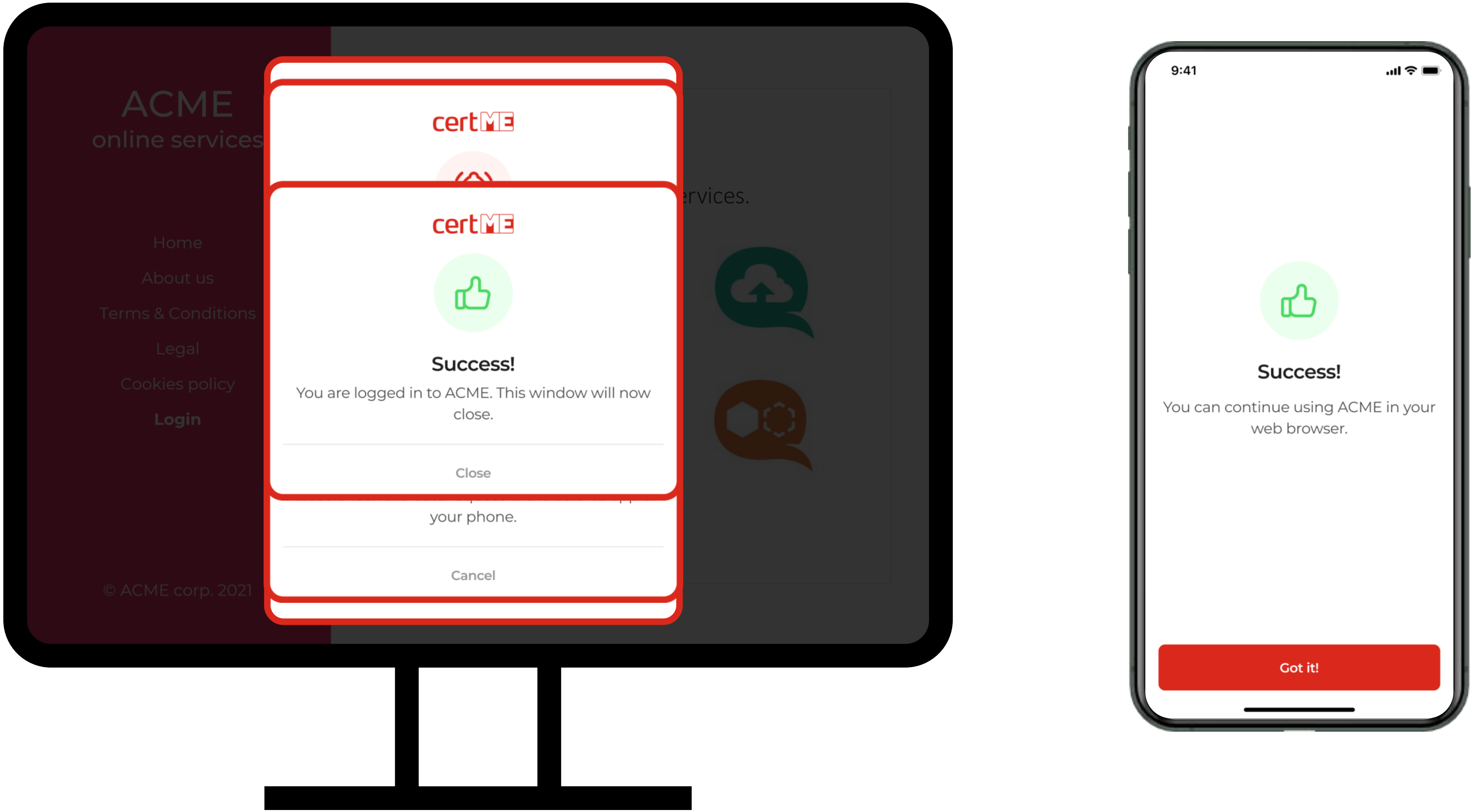
You can continue using ACME in your web browser.

[illegible]

certME

New customer registration

existing certME user becomes registered customer



certME

Success!

You are logged in to ACME. This window will now close.

Close

Success!

You can continue using ACME in your web browser.

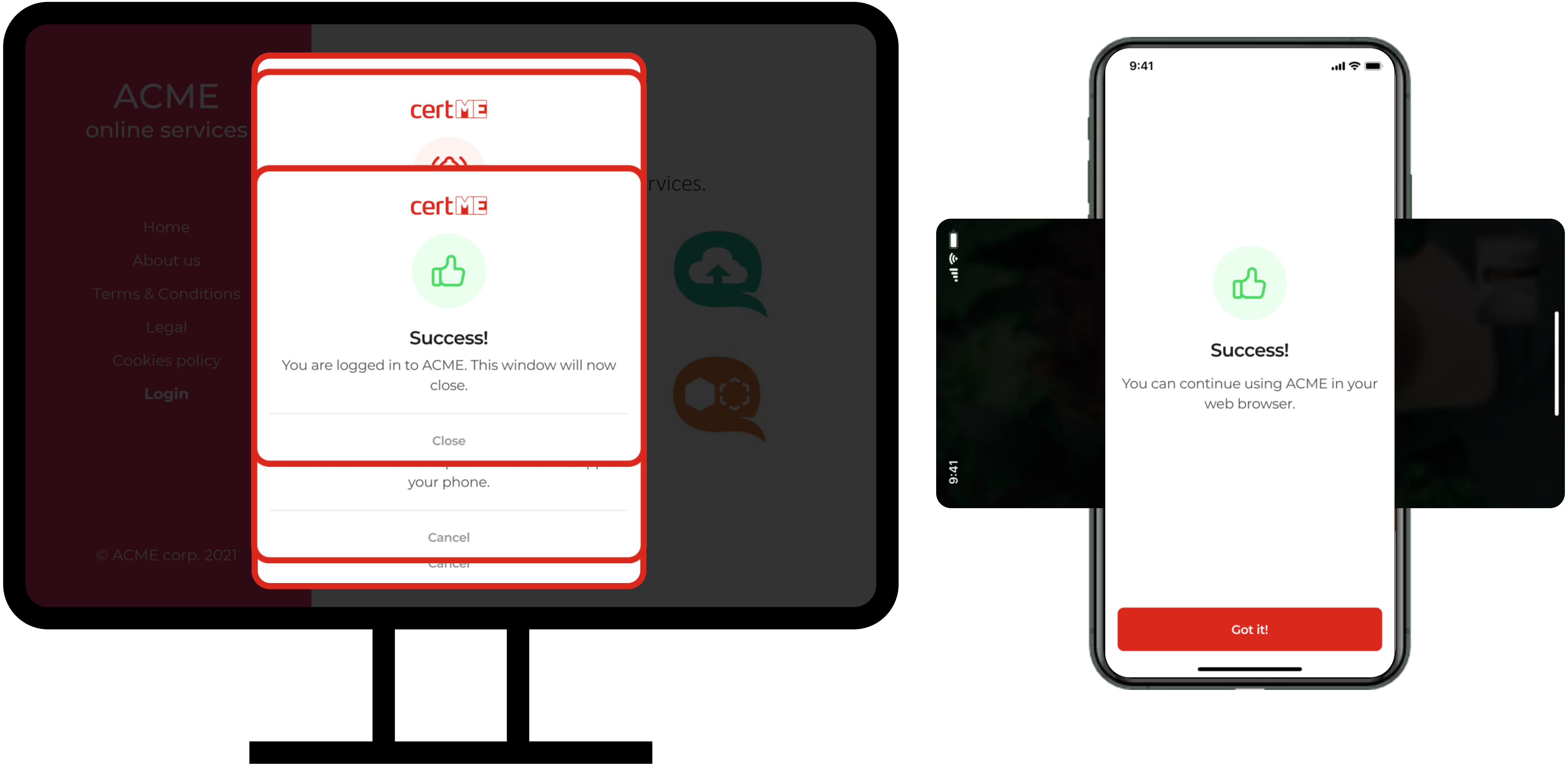
Got it!

When [illegible] account.

certME

Complete user journey

website visitor becomes certME user and registered customer



Success!

You are logged in to ACME. This window will now close.

your phone.

Success!

You can continue using ACME in your web browser.

When [illegible] account.

certME

Future work – IDBC project

Extending the EMI to a full fledged DIW

IDBC project

eID issuance based on previously performed verifications

Work with W3C VC/VP attestations

Attestation service compliant with OIDC VC

UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2014-2020

certSIGN, as a Beneficiary, in partnership with the University of Bucharest - Faculty of Mathematics, has been carrying out, starting with 14.10.2021, the project “Identity attestation services in decentralized environments based on blockchain technologies (**IDBC**)”.

The project is co-financed by the **ERDF** - European Regional Development Fund, through the Competitiveness Operational Program 2014-2020.

The content of this material does not necessarily represent the official position of the European Union or the Government of Romania

For more information

say **hello@certme.ro**