INTRODUCTION

👋 Overview of Civic Pass

Civic Pass is a powerful identity verification solution designed to enhance trust, control, and safety in digital interactions. It provides developers with a robust toolkit to implement user verification and access control in decentralized applications (dApps) and smart contracts.

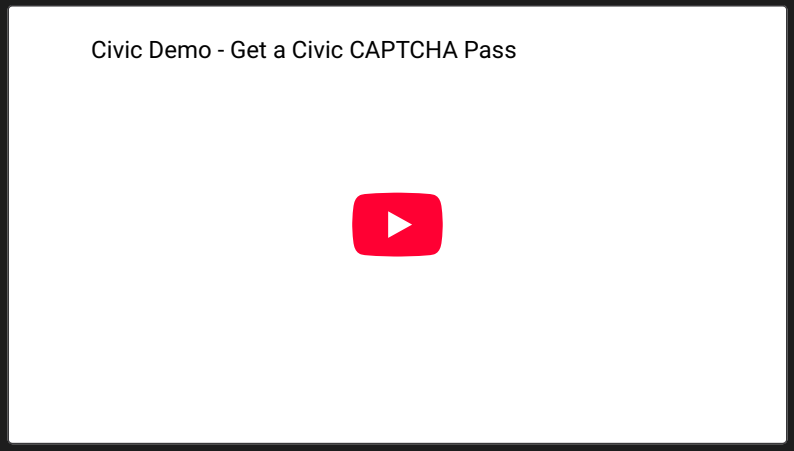
Key Features

- **On-chain attestation**: Civic Pass issues non-transferable tokens (soulbound or SBT) on the blockchain, serving as verifiable proof of a user's identity or attributes.
- **Flexible verification options**: Developers can configure Civic Pass to require various checks, including live video selfies, age verification, location confirmation, and ID document validation.
- **Multichain support**: Civic Pass is available on Solana and over 10 EVM-compatible chains, ensuring broad compatibility across blockchain ecosystems.
- **Privacy-focused**: Civic employs advanced security measures, including blockchain technology and encryption, to protect user information.

Standard Civic Passes

Builders can use standard Civic Passes off-the-shelf and benefit from the reusability of a verified address.

CAPTCHA Pass

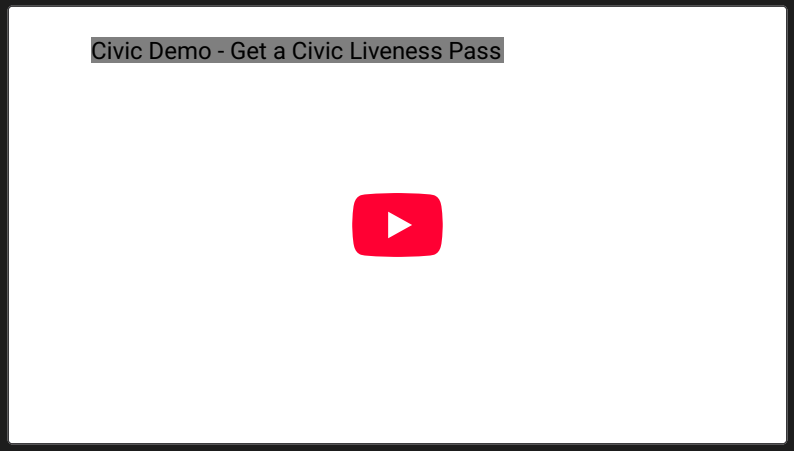


This pass requires a user to pass a challenge-response test to determine whether they are a human or bot.

Expiration and refresh required after 30 days.

Get CAPTCHA Key■

Liveness Pass

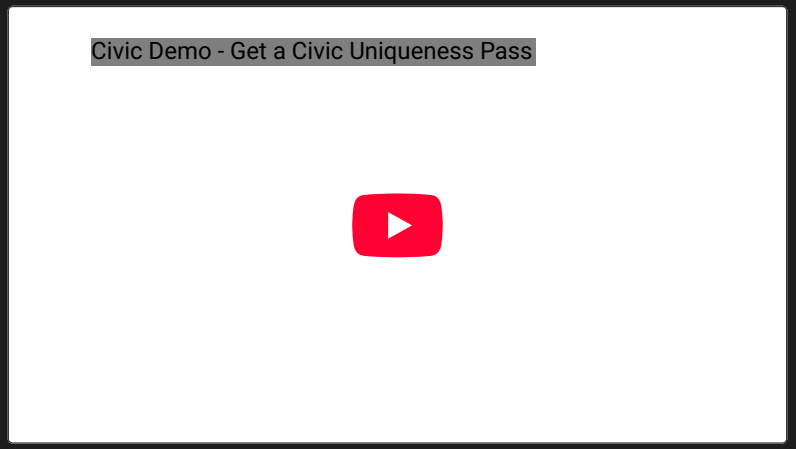


This pass requires a video selfie from a user to determine whether they are a human or bot. The pass will not be issued if VPN software is detected.

Expiration and refresh required after 30 days.

Get Liveness Key■

Uniqueness Pass



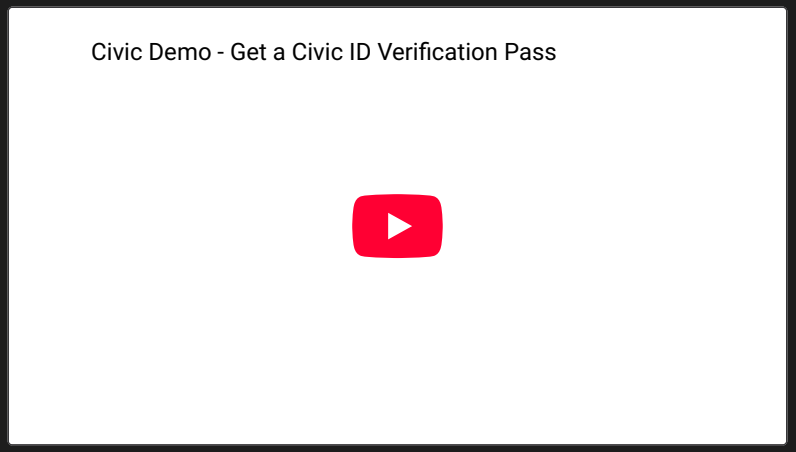
This pass requires a user to take a video selfie that also compares the face map resulting from the process with existing encrypted maps from the known universe of existing users. The pass will not be issued if VPN software is detected.

The known universe is confined to the network being used. A builder can request a private Uniqueness network by issuing a Custom Pass instead of the global one, when users are being offered embedded wallets where they don't control the private key.

Expiration and refresh required after 90 days.

Get Uniqueness Key■

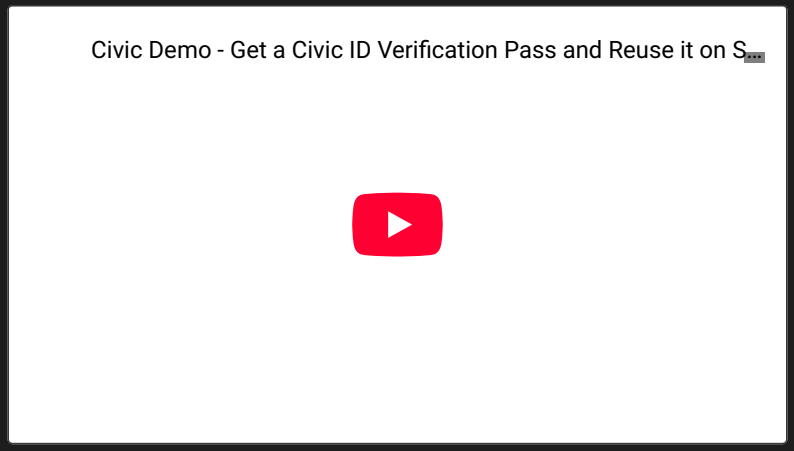
ID Verification Pass



This pass requires a user to pass a government-issued ID check. This check is combined with a liveness check, face comparison with the document, and a point-in-time sanctions check. A Uniqueness check can be added on-demand. The pass will not be issued if VPN software is detected.

This pass is privacy-preserving and does not contain user PII by default. Users can opt to store their data encrypted with a symmetric key for reusability, making the PII inaccessible without their consent. Builders **can** opt to retrieve underlying PII via a Civic-provided API endpoint. This is optional but recommended for projects that need to ensure adherence to KYC regulations.

Reusable Credentials Demo



These credentials are reusable until the earlier of the document expiration or the requirement for a refresh after 30 days.

Get in touch to discuss any ID verification requirements.

Custom Civic Passes

This pass allows businesses to tokenize their compliance on-chain via API. You can use your own ID verification provider. You can use Civic to tokenize the attestations of your verifications on-chain.

Custom Pass configurations include:

- Issue, freeze, revoke, and manage passes via API on a dedicated endpoint
- Use your own identity verification provider to issue passes
- Option to issue verifications to your users in abstracted wallets
- Option to issue passes on a private network with only your users
- Full lifecycle pass management and real-time analytics in your dashboard

Expiration and refresh cycle will depend on your business requirements.

Learn more about the functionality of the Custom Pass by accessing this section documenting the full OpenAPI specification.

Get in touch to discuss any custom business requirements.

Permissioned Web and Smart Contracts

Civic Pass enables the creation of permissioned environments in web applications and smart contracts:

- **Web applications**: Developers can gate access to specific pages or features, ensuring only verified users can interact with sensitive parts of the application.
- **Smart contracts**: By integrating Civic Pass checks, smart contracts can enforce rules based on user attributes, creating more secure and compliant on-chain interactions.

Implementation Considerations

- **User experience**: Always display the current status of a user's Civic Pass to maintain transparency.
- **VPN detection**: Civic Passes cannot be issued if a VPN is detected, ensuring the integrity of location-based verifications.
- **OFAC and other country restrictions**: See Supported Countries & Docs for the most updated list.

Previous
Quickstart

Next
Key Terms & Definitions

INTRODUCTION

📖 Key Terms & Definitions

Key Terms & Definitions

Civic Pass: A non-transferable token serving as an advanced identity and access management tool for both on-chain and off-chain applications. It enables users to verify their identity and access services securely across decentralized and traditional platforms.

Custom Civic Pass: A flexible, customizable identity solution that allows businesses to enforce compliance rules on-chain. Custom Civic Passes can be tailored to specific business needs using the Tenant Network.

Gateway Provider: A React component managing Civic Pass integration. It simplifies the process of integrating the Global Network, allowing Civic to handle user verification and issuance of the Civic Pass.

Gatekeeper Network: The network responsible for issuing and managing Civic Passes. Each Gatekeeper Network has a unique address, with specific rules and criteria for pass issuance and verification.

Network Key: A unique identifier that grants access to a specific Gatekeeper Network. This key is required to access Civic's Global Network and ensures that the rules of the Gatekeeper Network are followed during integration.

Attestation: A verified claim about a user's identity or attributes issued by a trusted party within the Gatekeeper Network. These claims are used to verify identity or other necessary information.

Verifiable Credentials: Cryptographically secured digital credentials that represent a user's attributes or qualifications. They can be used to verify identity or compliance without revealing sensitive personal information.

Verification Process: This includes CAPTCHA, Liveness, Uniqueness, and ID Verification, which confirm a user's identity according to set criteria. These checks ensure users meet the standards set by the Gatekeeper Provider and Gatekeeper Network for Civic Pass issuance.

Tokenized Identity: A digital representation of identity information stored on a blockchain, ensuring the integrity and security of identity attributes. Civic Pass serves as a tokenized identity that can be used across decentralized platforms.

On-chain/Off-chain: Refers to whether data or processes occur directly on the blockchain (on-chain) or through external systems (off-chain). Civic Pass can be used in both types of environments depending on the application's requirements.

Integration Concepts

React Component: This is the way you can trigger verification flows directly in your app. Projects using this method rely on Civic to handle the entire user verification and Civic Pass issuance process.

API: This method allows customers to directly manage the verification process and the issuance of Civic Passes to their users. This is only applicable to Custom Passes.

Fees & Costs

Blockchain Fees: Transaction costs associated with interacting with a blockchain network for pass issuance, verification, or other related processes.

Pass Fees: Costs associated with issuing and maintaining Civic Passes, including fees for verification and pass renewal. These fees depend on the

specific Gatekeeper Network or service provider.

User-Pay Model: In some cases, certain features or networks operate on a user-pays basis, where the user is responsible for covering the costs of verification.

Previous
Overview of Civic Pass

Next
Civic Pass Behavior

Last updated 7 months ago

Was this helpful?

INTRODUCTION

🔄 Civic Pass Behavior

Criteria for Issuing Civic Passes

All Passes

Users must meet **all** of the following criteria:

1. Agreed to Civic's Terms and Conditions, Privacy Policy, and Biometric Policy.
2. Completed wallet ownership verification by signing a transaction with their wallet.
3. Over the age of 18.
4. Not a resident or citizen of a blocked or banned location.
5. Not physically located in a blocked or banned location.

Blocked locations include: Bangladesh, China.

Banned countries: Afghanistan, Belarus, Burundi, Central African Republic, Cuba, Democratic Republic of Congo, Iran, Iraq, Lebanon, Libya, Myanmar, Nicaragua, North Korea, Russia, Somalia, South Sudan, Sudan, Syria, Venezuela, Yemen, Zimbabwe, and the following regions of Ukraine: Crimea/Sevastopol, Donetsk People's Republic, and the Luhansk People's Republic.

Liveness, Uniqueness, and ID Verification Passes

Users must meet **all** the above criteria and **additional** criteria below:

1. Agreed to Civic's Biometric Policy.
2. Completed VPN check. Using the IP address collected during the wallet ownership verification, Civic Pass will check the user's IP address to verify that they are not located in a blocked or banned location.
3. Completed a 3D FaceScan (video selfie of face).
   - The face will be analyzed by artificial intelligence technology to distinguish live humans from attempted spoofs. A topology of the face will be produced (FaceMap). The newly captured FaceMap of the user will be compared against the previously enrolled FaceMap to ensure the same user is using the wallet.
     - In the case of Uniqueness Passes and in some cases ID Verification, the captured FaceMap is stored and will be compared against any new FaceMaps incoming.

ID Verification Passes

Users must meet **all** the above criteria and **additional** criteria below:

1. Completed government-issued ID verification.

Identity data is collected and stored by Civic as stated in our Biometric Policy and in accordance with our Privacy Policy.

Civic Pass will only collect and store the following personal data of users requesting a Pass:

1. Email address
2. 3D FaceScan from a video selfie of face
3. Picture upload of government-issued ID
4. Data from ID document (combination of MRZ and OCR)
   - Name (full)
   - Date of birth
   - Document type
   - Document number

- Document expiration
- Country of issuance

Identity data verified by Civic:

Civic Pass will verify the following data on users requesting a Pass:

1. Email address verified with verification code
2. Government-issued ID authenticity
3. Selfie matches government-issued ID photo
4. Liveness

ID Verification Passes with PII Sharing

Users must meet **all** the above criteria and **additional** criteria below:

1. Agreed to share PII with the requesting project.

Customization Options for Custom Passes

Businesses will need API keys to access Custom Civic Passes. This integration method allows businesses to directly manage the issuance of Civic Passes to their users. Learn more about the functionality of the Custom Pass by reading the full OpenAPI specification.

Customization options include:

- Document type
- Age restrictions
- Country restrictions
- Pass expiration duration
- Combining Uniqueness with ID Verification
- Combining any features from different Pass types

Pass Statuses

After a Civic has been issued on-chain, the following status values may apply.

All Passes

Access Attempts from a Blocked Location

Attempts to access the app from a blocked location (e.g., China, Russia) will be blocked and the issued Pass will be frozen.

`Pass Status is FROZEN`

Access Attempts from a Banned Location

Attempts to access your app from a banned location (e.g., Cuba) will be blocked and the issued Pass will be revoked.

`Pass Status is REVOKED`

Pass Expiration

CAPTCHA and Liveness: 30 days

Uniqueness: 90 days

ID Verification: On the day of government-issued ID expiration, or one year after issuance, whichever comes first.

`Pass Status is EXPIRED`

Uniqueness Pass

Pass Refresh

Users will be required to refresh their Pass after it expires. Refreshing a Pass will happen automatically the next time the user connects to the requesting project. When connecting, Civic Pass will:

1. Request a user to sign a transaction with their wallet which will verify wallet ownership.
2. Check the user's IP address to verify that they are not located in a blocked or banned location.
3. Take a 3D FaceScan from a video selfie.
4. Match the FaceScan against the original FaceScan on the pass.

Successfully verifying wallet ownership and passing the above checks automatically reactivates a user's pass. An active Pass will remain active for 90 days after refresh unless it expires.

`Pass Status is REFRESHED / ACTIVE PASS or Pass Status is RECONNECTING / ACTIVE PASS`

Active Pass Early Pass Refresh

Each time user connects to your app with an Active Pass where the requesting project has forced a refresh, Civic Pass will:

1. Request a user to sign a transaction with their wallet which will verify wallet ownership.
2. Check the user's IP address to verify that they are not located in a blocked or banned location.
3. Re-verification depending on Pass type:
    a. **For CAPTCHA**: re-verify CAPTCHA.
    b. **For Liveness**: re-verify Liveness.
    c. **For Uniqueness or ID Verification**: Take a 3D FaceScan from a video selfie and match the FaceScan against the original FaceScan on the pass.

Successfully verifying wallet ownership and passing the above checks automatically reactivates a user's pass and updates the Pass expiration date.

`Pass Status is REFRESHED / ACTIVE PASS or Pass Status is RECONNECTING / ACTIVE PASS`

ID Verification Pass

Pass Refresh

Users will be required to refresh their Pass after it expires. Refreshing a Pass will happen automatically the next time the user connects to the requesting project. When connecting, Civic Pass will:

1. Request a user to sign a transaction with their wallet which will verify wallet ownership.
2. Check the user's IP address to verify that they are not located in a blocked or banned location.
3. Pass biometrics re-authentication.
4. Pass government-issued ID document verification.

Successfully verifying wallet ownership, passing the IP address check, passing government-issued ID verification, as well as the biometrics verification automatically refreshes a user's Pass. An active Pass will remain active for one year after refresh unless the document expires sooner.

`Pass Status is REFRESHED / ACTIVE PASS or Pass Status is RECONNECTING / ACTIVE PASS`

Reconnecting to Subscriber Property with Active Pass

Each time a user connects to your app with an Active Pass, Civic Pass will still check the user's IP address to verify that they are not located in a blocked or banned location.

When a user connects to your app with an Active Pass, Civic will check the Biometrics refresh status to ensure the same person is using the pass.

The Active Pass will remain active for one year after reconnecting or when document expires, whichever comes first. After one year, the pass will become inactive.

`Pass Status is REFRESHED / ACTIVE PASS or Pass Status is RECONNECTING / ACTIVE PASS`

PII Sharing

End user must consent to share PII with the Subscriber.

Subscriber has up to 24 hours to retrieve PII after End User consents to share. End user PII will no longer be available to you from Civic after 7 days or once retrieved, whichever comes first.

Identity data shared

With user's consent, Civic Pass will share the following data on individuals requesting a Pass:

1. Email Address
2. Data from ID document (combination of MRZ and OCR)
    - Name (full)
    - Date of birth
    - Document type
    - Document number
    - Document expiration
    - Country of issuance

Previous
Key Terms & Definitions
Next
Get Network Keys

Last updated 5 months ago

INTRODUCTION

🔑 Get Network Keys

Self-Serve Network Keys

This is the fastest way to start integrating Civic Pass. Request network keys for CAPTCHA, Liveness or Uniqueness Passes by filling out the appropriate form below. You will receive an automated email from Civic with a key to get started. See Pricing for more information on costs.

CAPTCHA Pass

A challenge-response test to determine human or bot. Try it out →

Get CAPTCHA key →

Liveness Pass

A video selfie to determine human or bot. Try it out →

Get Liveness key →

Uniqueness Pass

A video selfie to determine 1-user-1-wallet. Try it out →

Get Uniqueness key →

Network Keys That Require a Contract

Our team will be in touch to learn more about your business needs. See Pricing for more information on costs.

ID Verification Pass

Verifies real-world identity using government-issued ID documents. Try it out →

Get in touch →

Custom Pass

Need additional countries, on-chain checks, not yet supported chains, or other verifications?

Get in touch →

What are Network Keys?

Network Keys are unique identifiers that reference a specific Civic Pass type, such as a CAPTCHA Pass or a Liveness Pass. They are used to identify a particular pass type within the Civic system, though they may not always be unique to a given application. However, passes that have custom requirements will be issued unique network keys for the specific pass.

Network Keys are IDs that correspond to a specific pass type on the Civic side. They are used to identify the type of pass being utilized and help query whether users possess a certain pass type. They do not serve as authentication credentials for applications to access the Civic Pass system.

For example, builders that want to ensure user uniqueness can rely on the Uniqueness Pass network and use its corresponding network key to query the on-chain state of the pass in each of the user wallets presented in their application. They cannot issue a pass directly but can call for passes to be issued through the Civic SDKs, using the network key to identify the pass types.

Testing with Network Keys

You can use a network key on testnet/devnet to test your integration with Civic Pass. This allows you to simulate real-world scenarios and verify that your application is correctly configured.

Key Points

- **Not application-specific**: Network Keys are tied to pass types, not specific applications.
- **No secure communication implication**: Network Keys do not imply or facilitate secure communication through encryption. They simply allow on-chain identification of the pass being used.
- **No authentication role**: They do not authenticate applications to the Civic system; front-end applications do not require authentication to use them.
- **No data encryption**: Network Keys do not encrypt or decrypt data transmitted between applications and the Civic Pass system.

Troubleshooting

If you encounter issues such as authentication errors or invalid key messages:

1. **Verify key accuracy**: Double-check that you have entered the correct Network Key and that it matches the one provided by Civic Pass.

2. **Contact support**: Reach out to the Civic team on Discord.

Previous
Civic Pass Behavior

Next
Referral Link Guide

Civic Docs

INTRODUCTION

🔗 Referral Link Guide

How any partner can embed a pre-configured Civic Pass URL in their product or campaign

1. Why use a pre-configured link?

Embedding the right query parameters up-front means users arrive with exactly the **Pass type**, **blockchain**, and **partner attribution** you need, so:

Benefit	Detail
Friction-free UX	Users skip drop-downs and get straight to verification.
Accurate attribution	The `referrer` tag lets you track traffic and revenue-share precisely.
Zero-code updates	Adding new Pass scopes or chains is as simple as editing the URL.

2. Referral-link anatomy

```
https://getpass.civic.com/
  ?scope=<scopes>          # Pass scopes, comma-separated
  &chain=<chains>          # Allowed blockchains, comma-separated
  &referrer=<your-slug>    # Partner identifier (lowercase,
URL-safe)
```

Parameter			
`scope`	✅	`uniqueness`	Pass scope(s) to issue.
`chain`	✅	`base,ethereum`	Restricts issuance to the listed networks.
`referrer`	✅	`partner-xyz`	Tags verifications for analytics &

> **Tip:** Spaces in chain names must be URL-encoded (`%20`). For example, `arbitrum one` → `arbitrum%20one`.

3. Building your link

1. **Pick scopes** – choose 1-N from `uniqueness`, `captcha`, `liveness`, `kyc`, ...
2. **Pick chains** – list any EVM chain slug (`base`, `optimism`, `polygon`, ...).
3. **Get your referrer slug** – request it from your Civic account manager.

```
Example
--------
Scope(s):  uniqueness
Chain(s):  base
Referrer:  partner-xyz

URL:
https://getpass.civic.com/?
scope=uniqueness&chain=base&referrer=partner-xyz
```

Feel free to experiment in a browser; the link is self-validating.

4. How to embed the link

4.1 Static HTML - simplest

```
<a href="https:
scope=uniquenes
   target="_bla
  Verify with C
</a>
```

This site uses cookies to deliver its service and to analyze traffic. By browsing this site, you accept the privacy policy.

4.2 React (ES5 class)

```
var CivicPassButton = React.createClass({
  render: function () {
    var url = 'https://getpass.civic.com/?
scope=uniqueness&chain=base&referrer=partner-xyz';
    return React.createElement(
      'a',
      { href: url, target: '_blank', rel: 'noopener noreferrer',
className: 'btn btn-primary' },
      'Verify with Civic Pass'
    );
  }
});
```

4.3 Next.js / Link component

```
import Link from 'next/link';

export default function CivicPassLink() {
  const url =
    'https://getpass.civic.com/?
scope=uniqueness&chain=solana&referrer=partner-xyz';
  return (
    <Link href={url} target="_blank" rel="noopener noreferrer">
      Verify with Civic Pass
    </Link>
  );
}
```

5. QA checklist before going live

Open the link in a private/incognito window; confirm only your chosen scopes appear.

Switch your wallet to a non-allowed chain and confirm issuance is blocked.

Complete a test ver
`referrer` slug.

If using `redirect_`

6. Troubleshooting guide

Symptom	Likely cause	Fix
Extra scopes showing	Wrong `scope=` list	Keep only the scopes you need
Users not restricted to a single chain	Multiple chain slugs present	Limit `chain=` to one entry
Analytics show “unknown” referrer	Typo or missing `referrer=`	Confirm slug with Civic team

7. Frequently asked questions

Q. Can I support multiple chains?
Yes—comma-separate them: `chain=base,optimism,polygon`.

Q. How do I support testnets?
Use the same URL; Civic follows the wallet network. Just ensure your dApp is on the matching test chain (e.g., Base Sepolia).

Q. What if I add CAPTCHA later?
Update the link to `scope=uniqueness,captcha`. No code changes beyond that.

Previous
Get Network Keys

Next
Internet Computer (ICP)

Last updated 18 days