Building Privacy-Preserving Cryptographic Credentials from Federated Online Identities

John Maheswaran
PhD Dissertation Defense, 6/24/2015
Department of Computer Science
Yale University

Committee:
Bryan Ford (adviser)
Joan Feigenbaum
Ramakrishna Gummadi
Anil Somayaji (Carleton University)

Crypto-Book

Roadmap

1. Background

2. Work Overview

3. System Architecture

4. Credential Producers and Consumers
   - At -Large Credentials
   - Group Credentials

5. Evaluation

6. Conclusions

Roadmap

1. **Background**

2. Work Overview

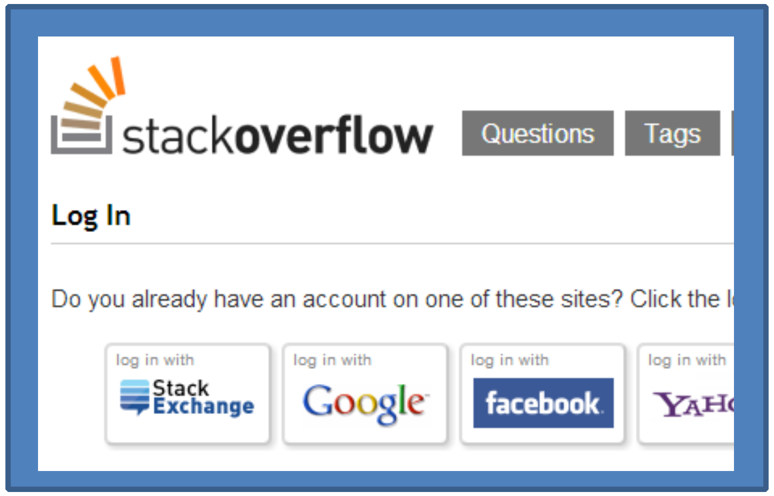
3. System Architecture

4. Credential Producers and Consumers
   - At -Large Credentials
   - Group Credentials

5. Evaluation

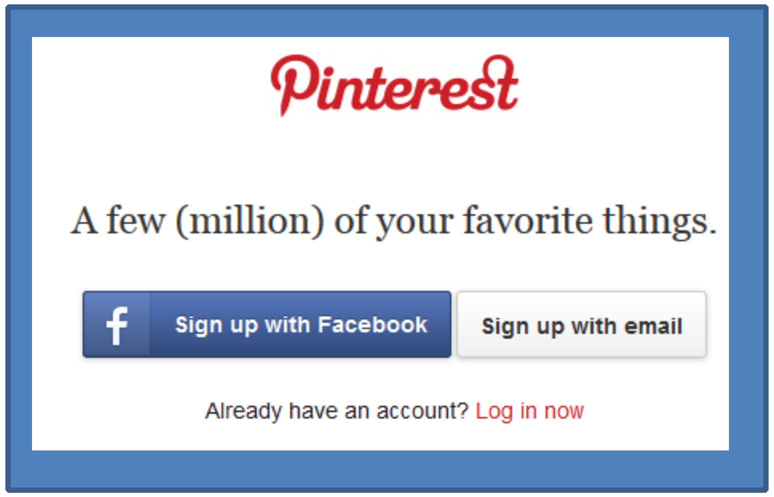
6. Conclusions

Background: Federated Authentication



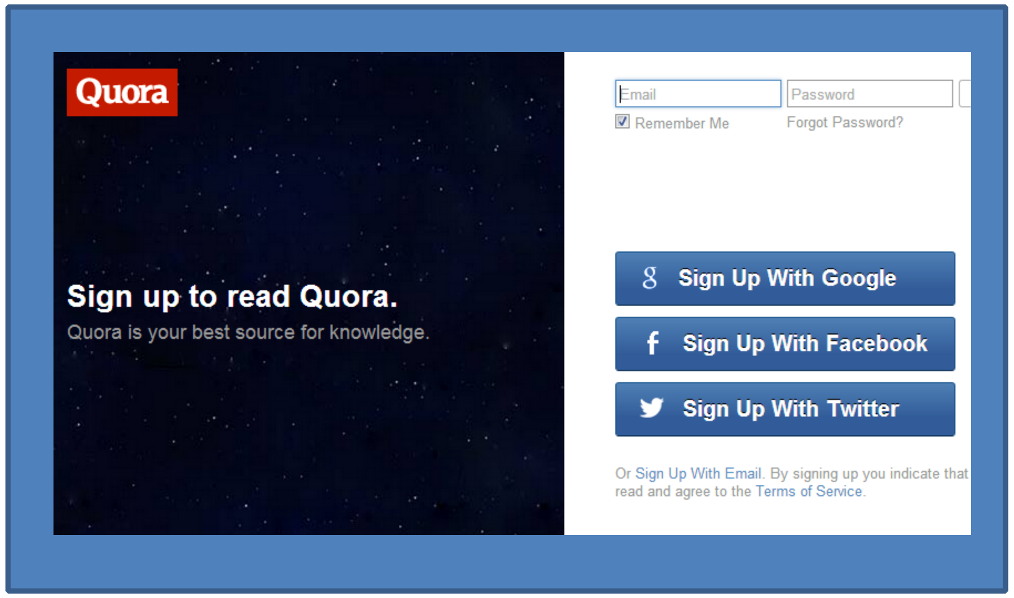
Log In

Do you already have an account on one of these sites? Click the l



A few (million) of your favorite things.

Already have an account? Log in now

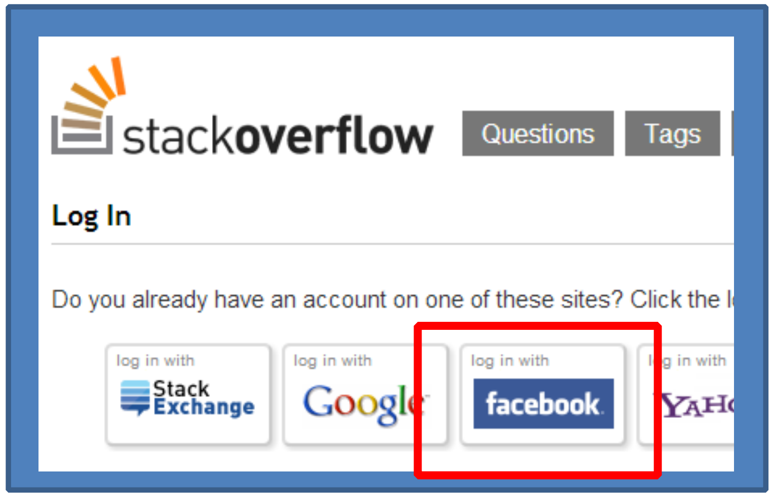


Sign up to read Quora.

Quora is your best source for knowledge.

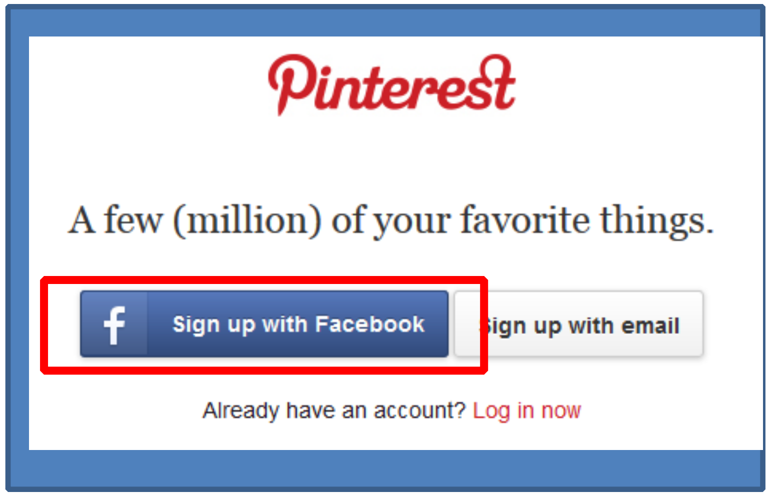
Or Sign Up With Email. By signing up you indicate that read and agree to the Terms of Service.

Background: Federated Authentication



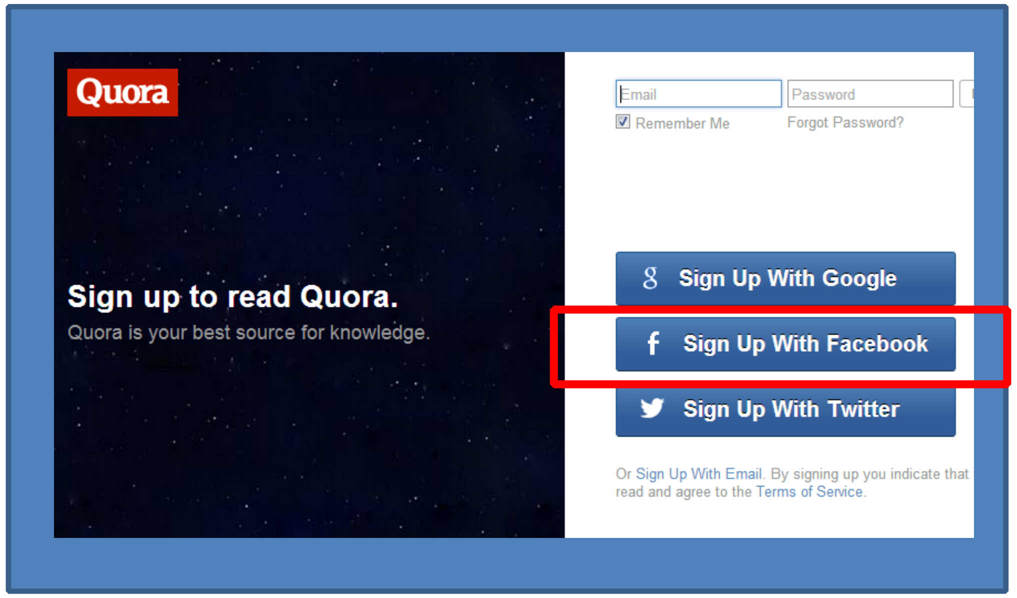
Log In

Do you already have an account on one of these sites? Click the l



A few (million) of your favorite things.

Already have an account? Log in now

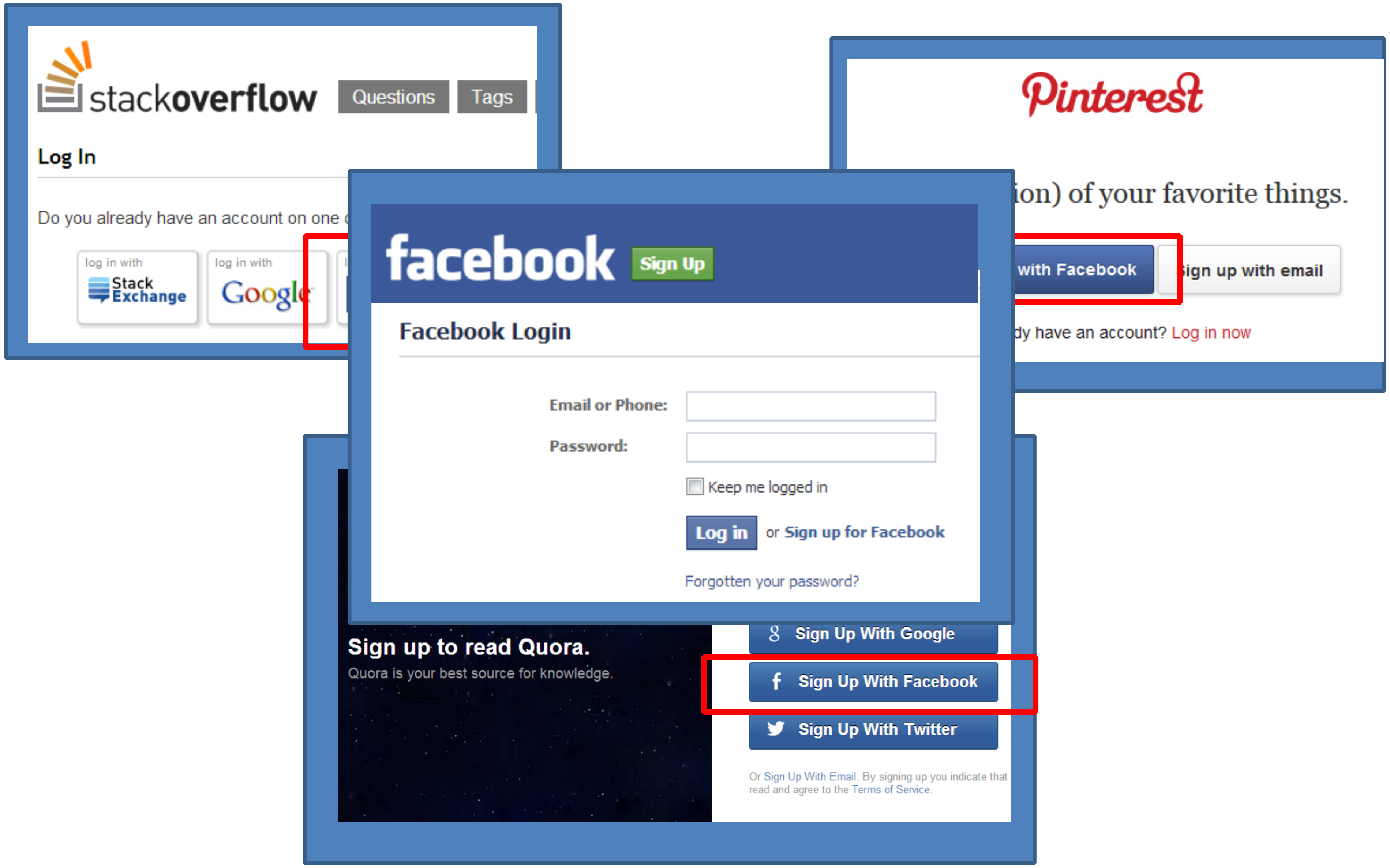


Sign up to read Quora.

Quora is your best source for knowledge.

Or Sign Up With Email. By signing up you indicate that read and agree to the Terms of Service.

Background: Federated Authentication



Do you already have an account on one

ion) of your favorite things.

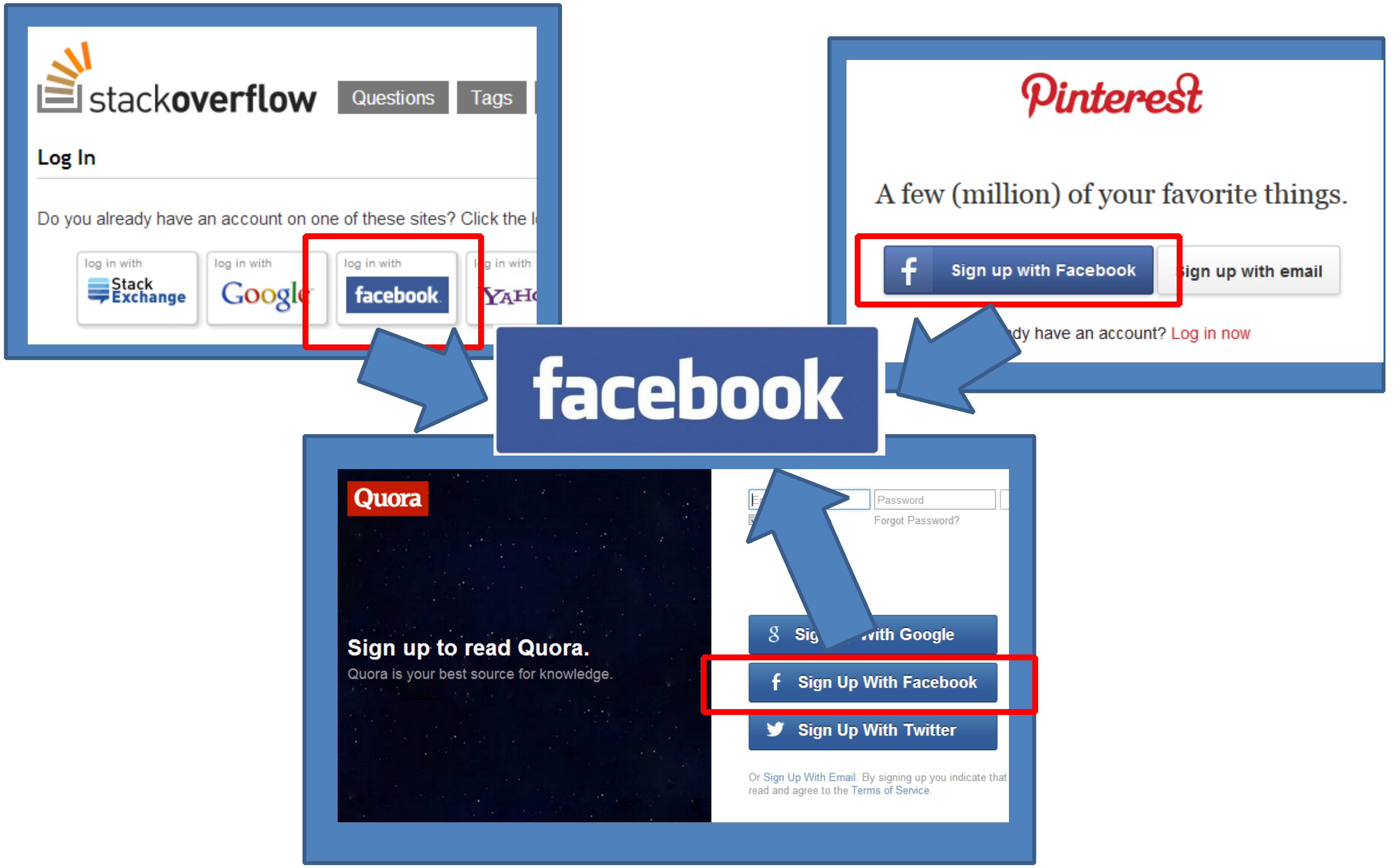
dy have an account? Log in now

Sign up to read Quora.

Quora is your best source for knowledge.

Or Sign Up With Email. By signing up you indicate that read and agree to the Terms of Service.

Background: Federated Authentication



Do you already have an account on one of these sites? Click the l

A few (million) of your favorite things.

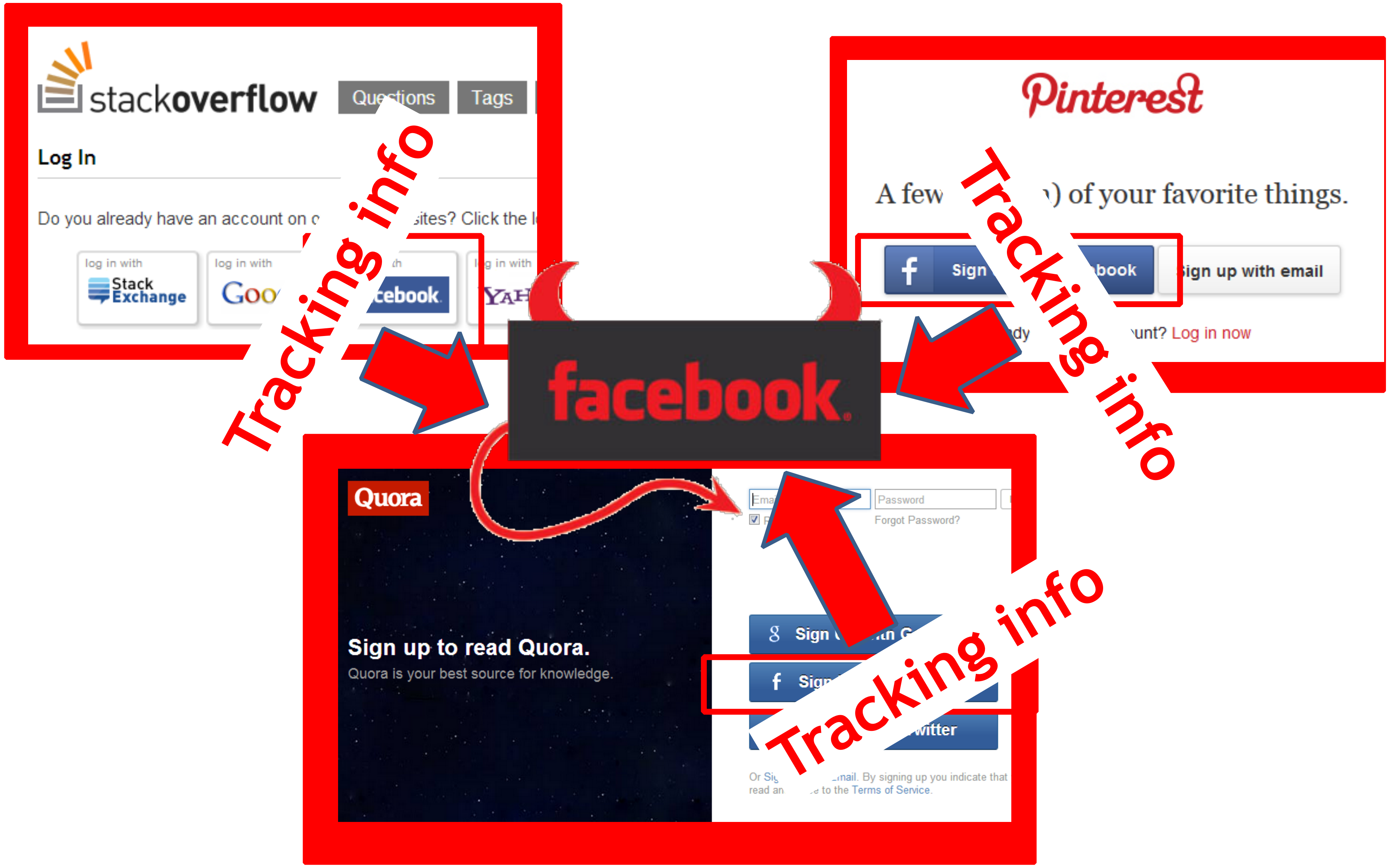
dy have an account? Log in now

Sign up to read Quora.

Quora is your best source for knowledge.

Or Sign Up With Email. By signing up you indicate that read and agree to the Terms of Service.

Background: Federated Authentication



Background: Federated Authentication

- Popular for managing online identities
- Examples: Facebook and PayPal
- Authentication protocols such as OpenID/OAuth
- Privacy cost: ID provider and applications can track users across all sites

Federated Authentication **Privacy Concerns**

- ID providers learns every application user logs into
- ID providers learns login time to every application for a user
- ID provider can impersonate user on applications
- Applications learn the user’s true identity
- Applications learn user profile details e.g. friends lists, location

Federated Authentication **Privacy Concerns**

- Applications can edit user profile on ID provider e.g. post to timeline, edit personal info
- Applications can link user behavior across sites
- User data can be tracked and sold to advertisers
- Compromised federated ID account can log in as that user to all applications

Motivating Use Case: Wikipedia Anonymous Editing

- Privacy preserving login to Wikipedia
- In favor of anonymous editing
- Anonymous editing often abused - vandalism/spam
- Anonymous yet abuse resistant editing
- Allow users to edit pages without revealing their identities
- Allow admins to sanction site abusers

Motivating Use Case: Group Authenticated SecureDrop

- Verifiable whistleblowing without compromising privacy
- Allow a journalist to authenticate leaked documents without compromising source anonymity
- A whistleblower authenticates as a member of a group and signs document
- Journalist knows that the document came from a director at Evil Corp. Inc. but does not know which one

Related Work

- PseudoID Dey and Weis. [HotPets ’10]
  - privacy protected federated login
  - does not handle key assignment or Sybil resistance
- Location privacy via private proximity testing Narayanan et al. [NDSS ’11]
  - Proposed using social network as a PKI
- Opaak Maganis et al. [MobiSys ’12]
  - provides Sybil resistance by relying on a cellphone as scare resource.
- SudoWeb Kontaxis et al. [Information Security 2011]
  - looked at limiting the amount of Facebook information disclosed to third party sites
  - did not consider anonymous online IDs

Roadmap

1. Background
2. **Work Overview**
3. System Architecture
4. Credential Producers and Consumers
    - At -Large Credentials
    - Group Credentials
5. Evaluation
6. Conclusions

Work Overview

- [Poster] Crypto-Book: Privacy Preserving Online Identities; **John Maheswaran**, David Isaac Wolinsky, Bryan Ford; SOSP '13 Poster Session (Symposium on Operating Systems Principles); and Diversity '13 Poster Session (Workshop on Diversity in Systems Research)

- [Extended abstract/WIP] Crypto-Book: Privacy Preserving Online Identities; **John Maheswaran**, David Isaac Wolinsky, Bryan Ford; SOSP '13 Works In Progress (WIP) Session (Symposium on Operating Systems Principles)

- [Paper] Crypto-Book: An Architecture for Privacy Preserving Online Identities; **John Maheswaran**, David Isaac Wolinsky, Bryan Ford; HotNets ’13 (Hot Topics in Networks ’13)

Work Overview

- [arXiv tech report] Crypto-Book: Bootstrapping Privacy Preserving Online Identities from Social Networks; **John Maheswaran**, Daniel Jackowitz, David Isaac Wolinsky, Lining Wang, Bryan Ford arXiv preprint arXiv:1406.4053, June 2014

- [Paper *(under submission)*] Building Privacy-Preserving Cryptographic Credentials from Federated Online Identities; **John Maheswaran**, Daniel Jackowitz, Ennan Zhai, David Isaac Wolinsky, Bryan Ford; CoNEXT '15 (ACM Conference on emerging Networking Experiments and Technologies)

Press coverage

- The workshop on diversity in systems research 2013; Christopher Stewart and Vishakha Gupta; **ACM SIGOPS Operating Systems Review** 48.1 (2014): 103-106.
- The federation of our digital identities; **Is Nerd Science blog**; http://isnerd.co/2014/07/05/federated-identity-privacy-namecoin/
- CryptoBook; **Layer 9 Computer networking and systems research blog**; http://www.layer9.org/2013/11/hotnets-13-cryptobook.html

Online resources

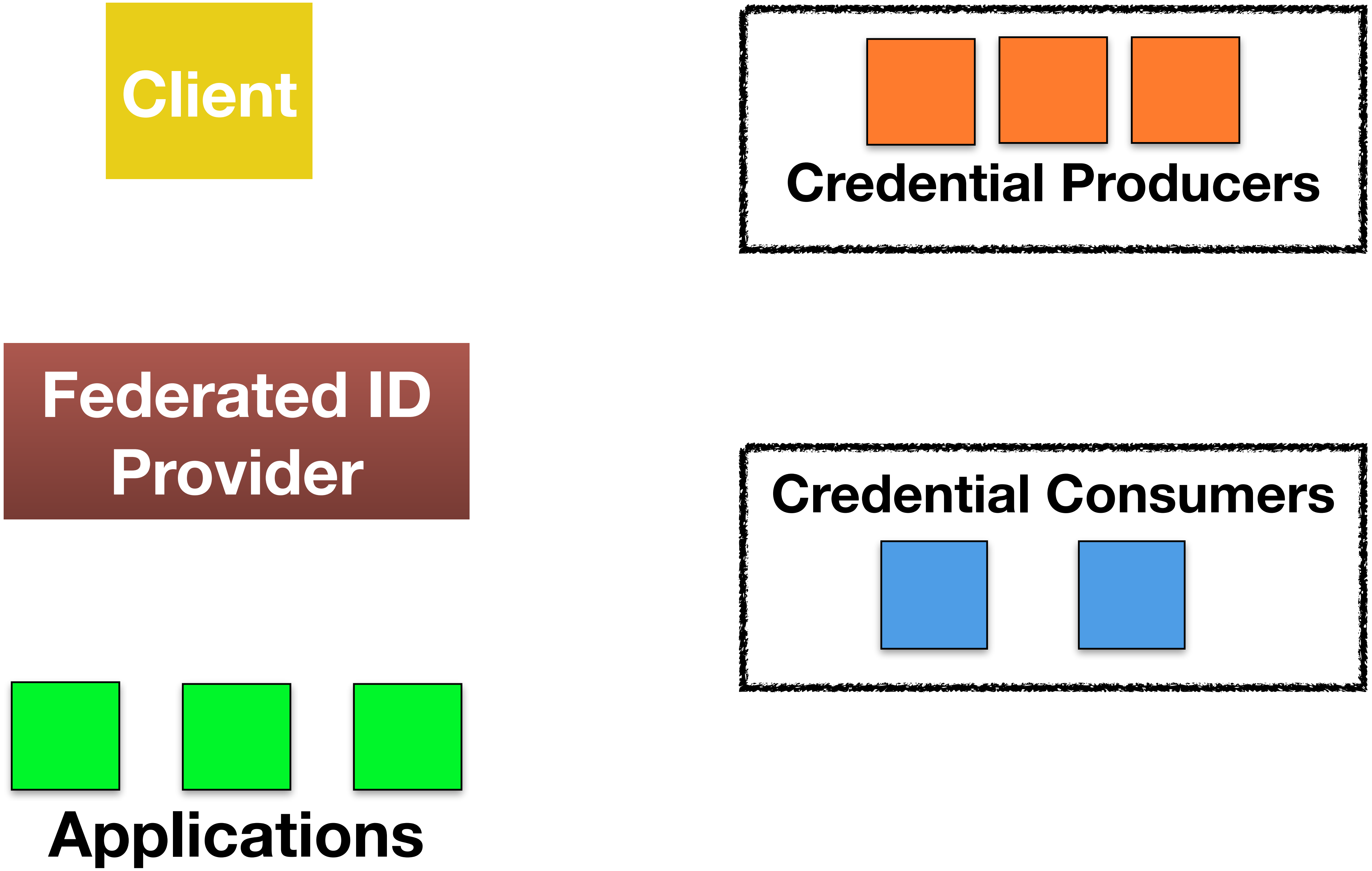
- Open source code is available on GitHub:
  - github.com/jyale/cobra
- Project websites:
  - www.crypto-book.com
  - www.cryptobook.ninja

Crypto-Book

Roadmap

1. Background
2. Work Overview
3. **System Architecture**
4. Credential Producers and Consumers
    - At -Large Credentials
    - Group Credentials
5. Evaluation
6. Conclusions

System Components



Client

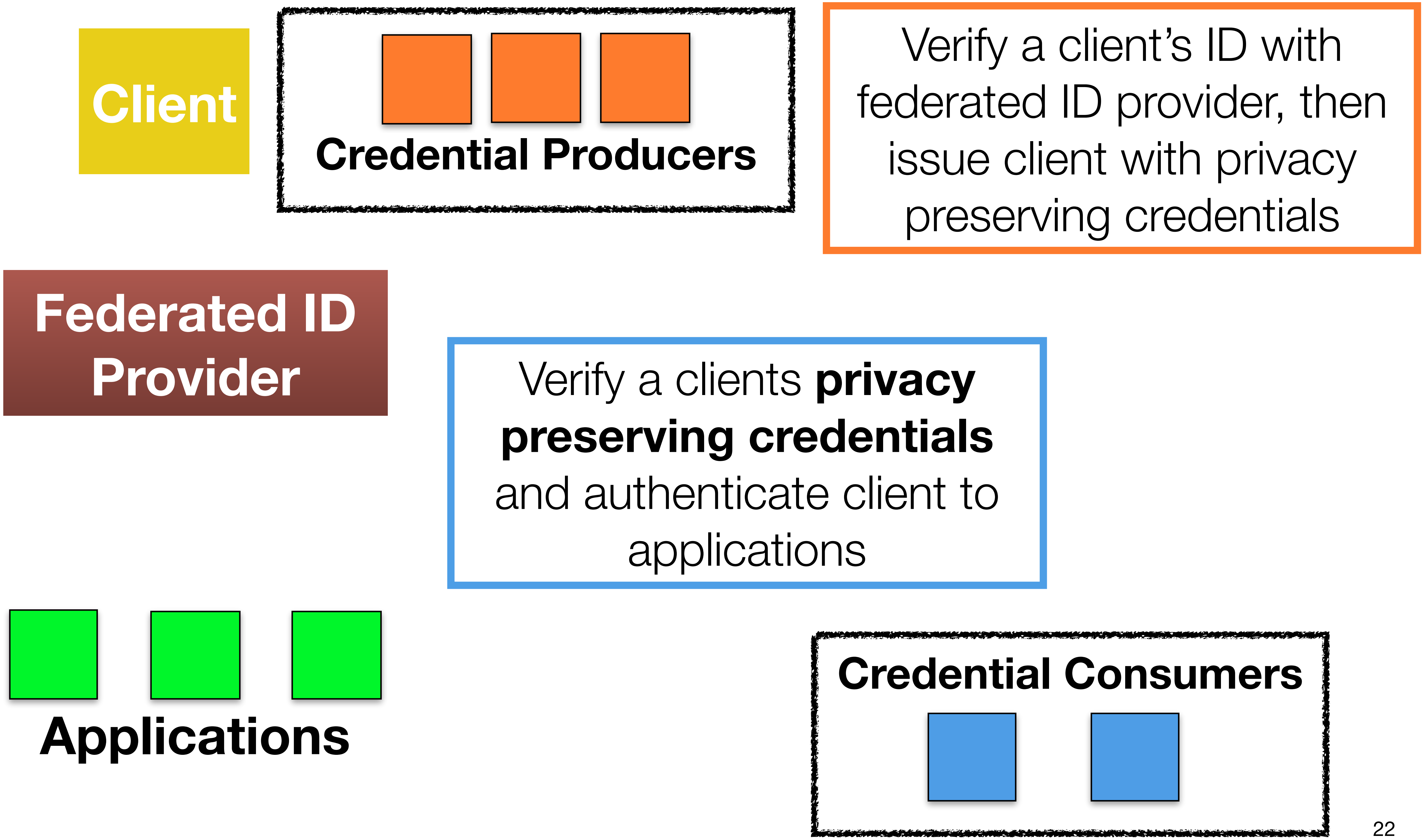
Federated ID Provider

Applications

Credential Producers

Credential Consumers

System Components



Client

Credential Producers

Verify a client’s ID with federated ID provider, then issue client with privacy preserving credentials

Federated ID Provider

Verify a clients **privacy preserving credentials** and authenticate client to applications

Applications

Credential Consumers

Security Properties

- **Anonymity** No single party can unmask a pseudonym to a federated ID
- **Unlinkability** It is not possible to tell if two pseudonyms are controlled by the same person
- **Accountability** (abuse resistance) A user can be punished if they misbehave (e.g. spam/troll)
- **Unforgeability** (no impersonation) No one can act as the user and authenticate as them

Threat Model: **Threats**

- **Clients** post low quality content/spam
- **Federated ID providers and applications**
  - de-anonymize client
  - learn what applications client accesses
- **Multiple applications** link client's identity across sites

Threat Model: **Assumptions**

- At most (t-1) of n credential producers are dishonest
  Others are honest-but-curious.

- Do not consider network level attacks
  Clients can connect to system components via anonymous networks (e.g. Tor)

- Anonymous network communication/cryptographic primitive compromise are outside of scope

Client

- Person browsing the web
- Interacts with other system components via browser
- Interacts with all other components in system
- Goal is to login to and use a web application

Client

Application

Application

- A web site that someone wants to use
- Client authenticates to log in to their account on that website
- Many applications now support federated authentication (e.g. Log in with Facebook/Log in with LinkedIn etc)
- Examples:

Quora

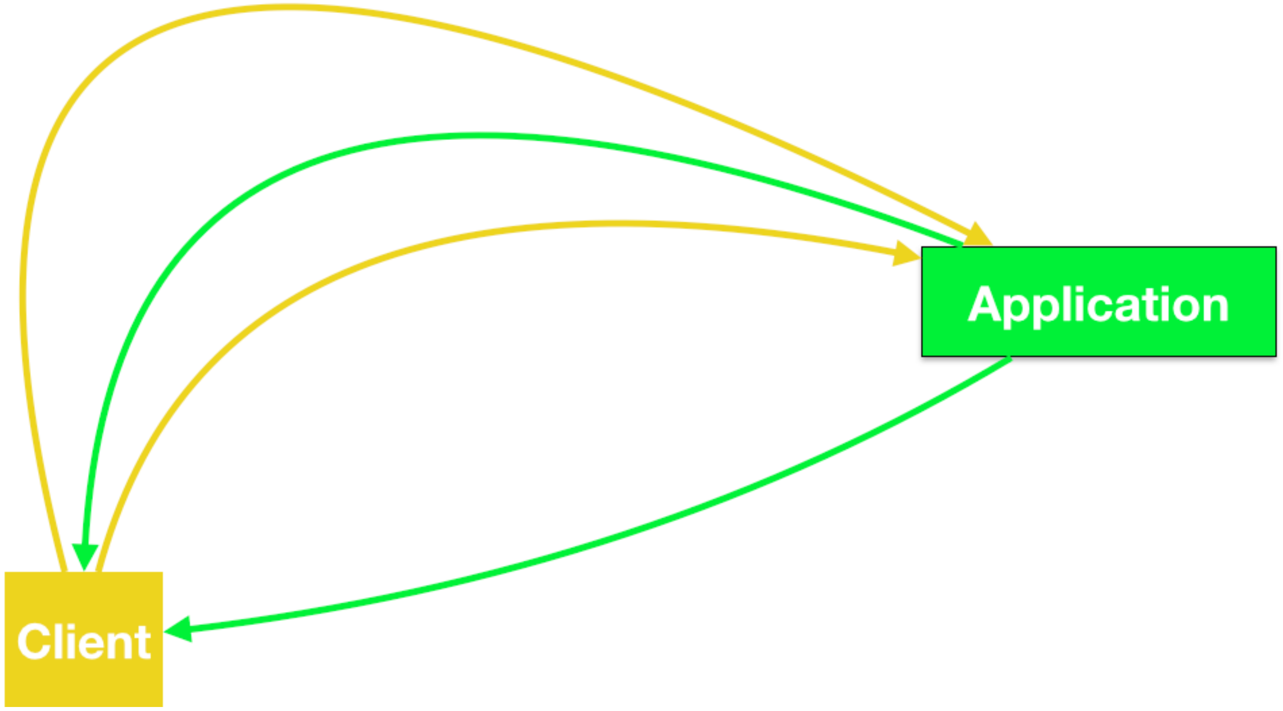
Pinterest

stackoverflow

slideshare

venmo

Non-federated client-application interaction

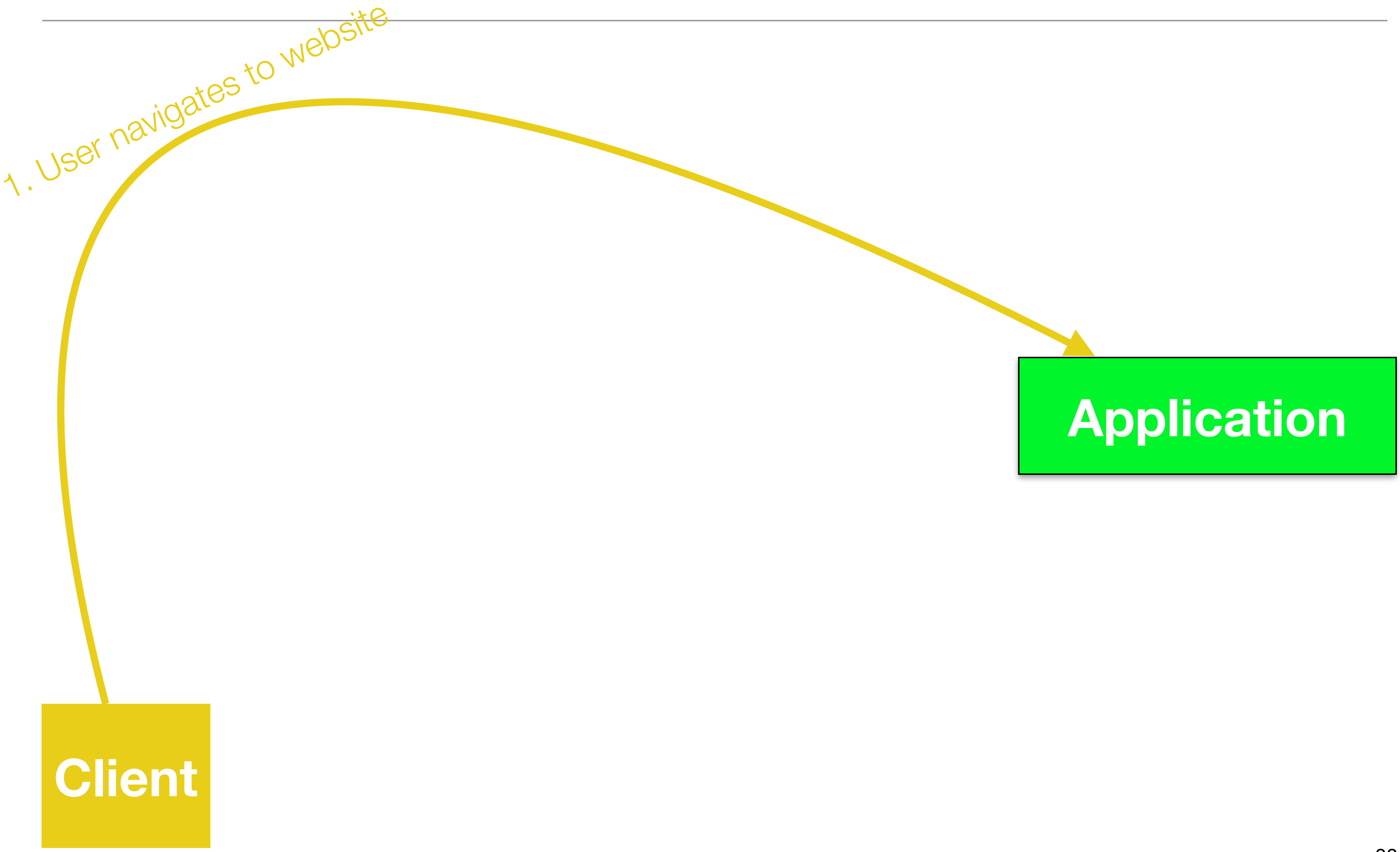


Non-federated Client-Application interaction

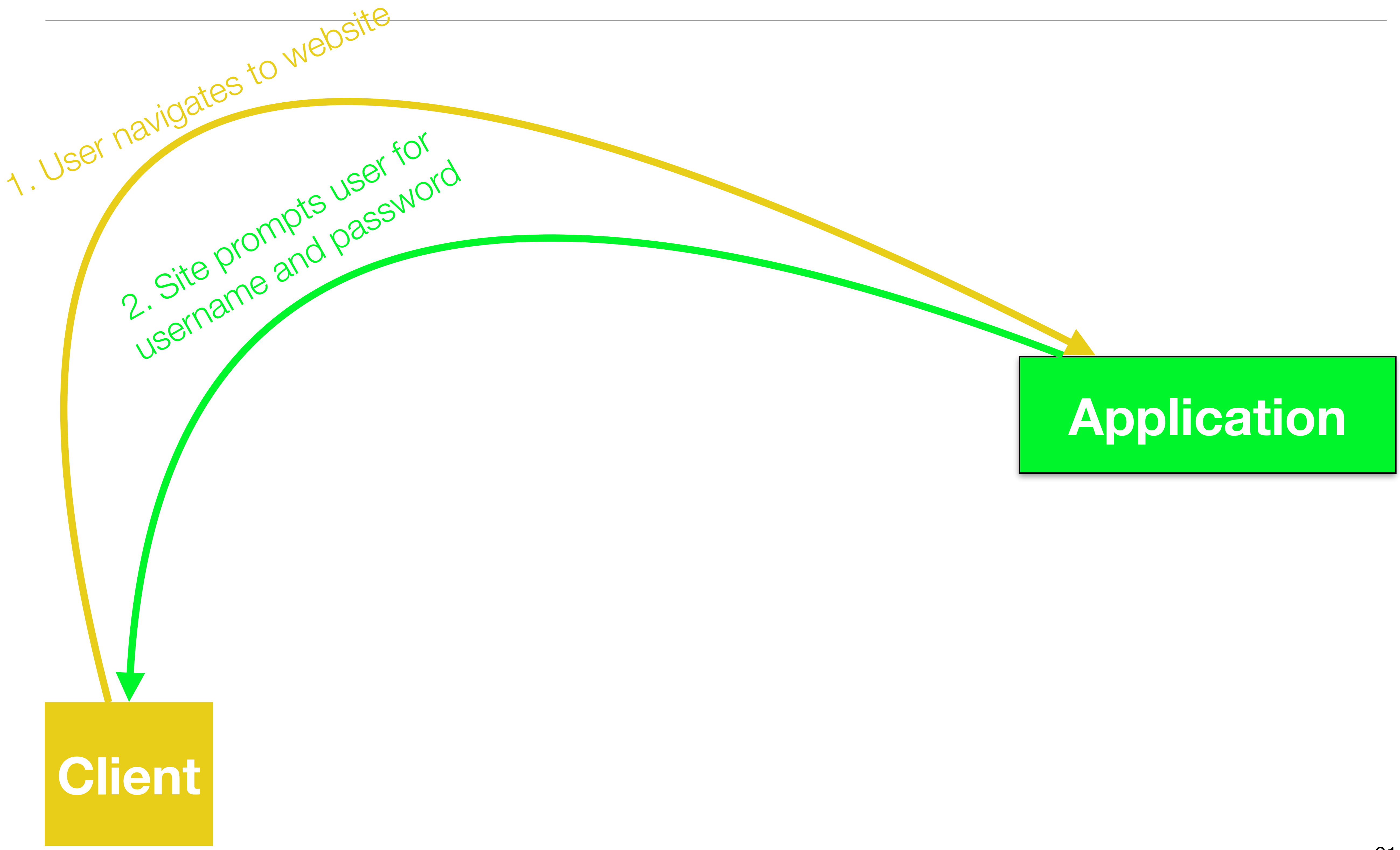
Application

Client

Non-federated Client-Application interaction

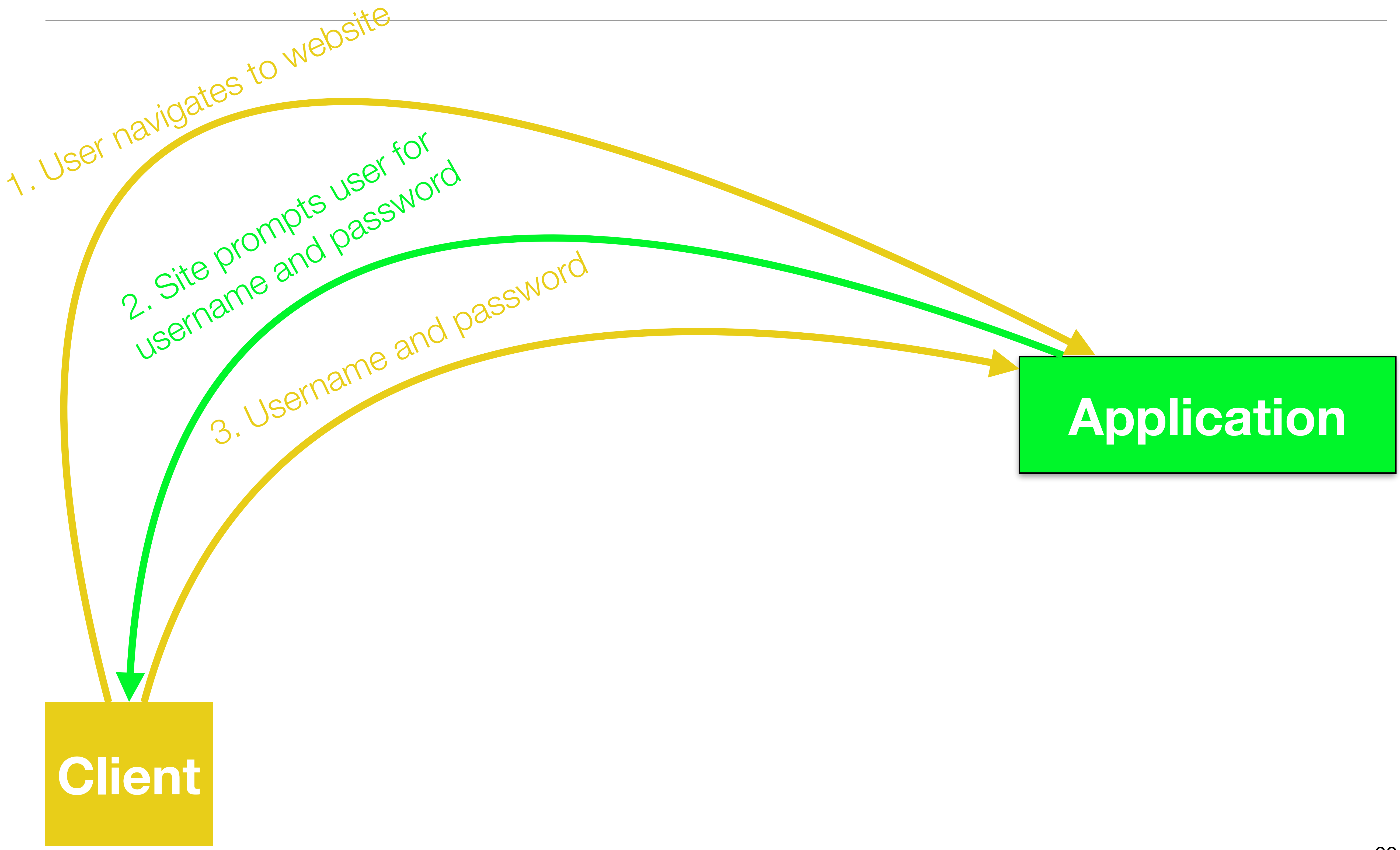


Non-federated Client-Application interaction



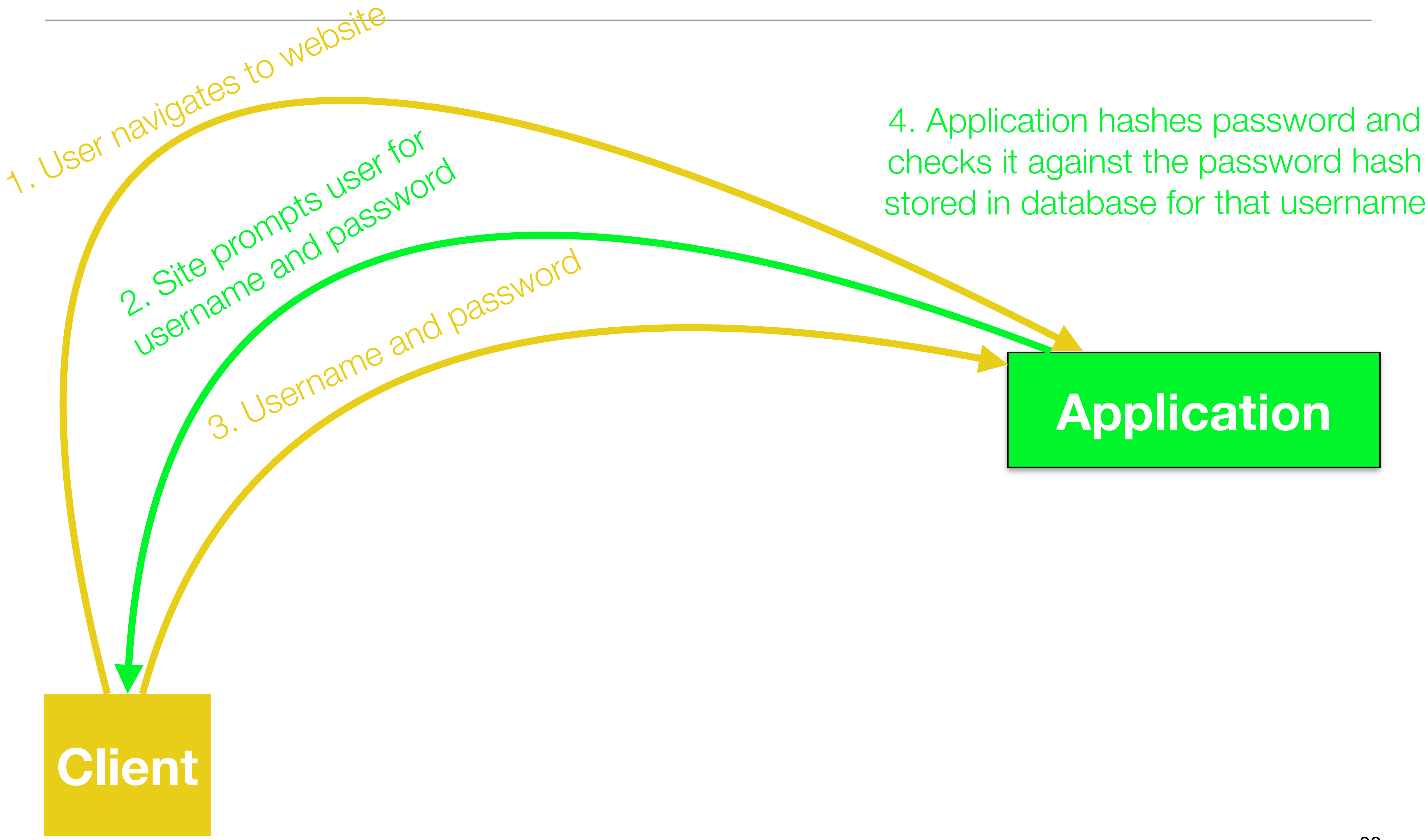
1. User navigates to website
2. Site prompts user for username and password
Application
Client

Non-federated Client-Application interaction



1. User navigates to website
2. Site prompts user for username and password
3. Username and password
Application
Client

Non-federated Client-Application interaction



1. User navigates to website

2. Site prompts user for username and password

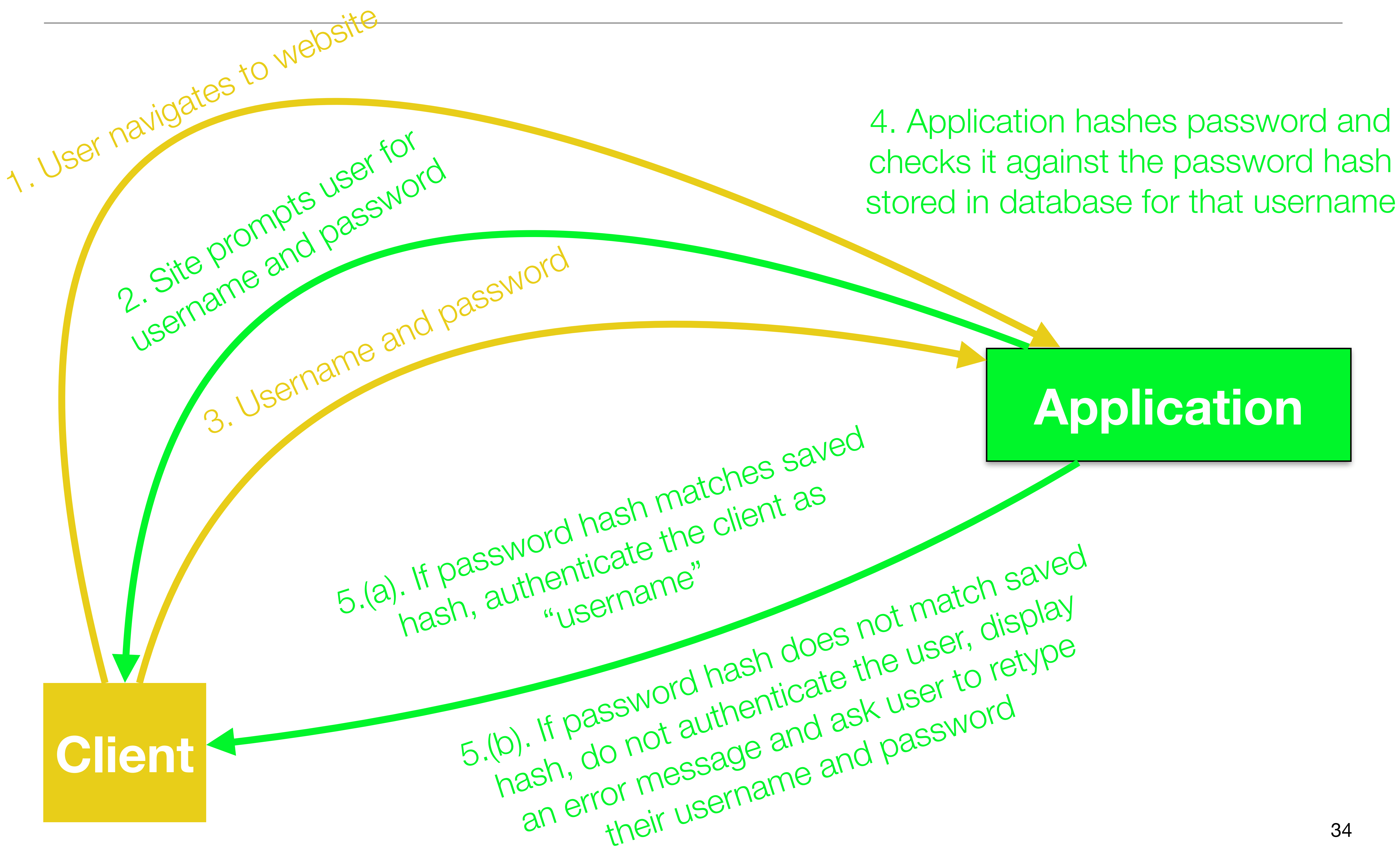
3. Username and password

4. Application hashes password and checks it against the password hash stored in database for that username

Application

Client

Non-federated Client-Application interaction



1. User navigates to website

2. Site prompts user for username and password

3. Username and password

4. Application hashes password and checks it against the password hash stored in database for that username

5.(a). If password hash matches saved hash, authenticate the client as “username”

5.(b). If password hash does not match saved hash, do not authenticate the user, display an error message and ask user to retype their username and password

Federated Identity Provider

Federated ID Provider

- Authenticates users for applications
- Often a social network or other identity provider
- Financial ID providers (e.g. PayPal) require real world verification - Higher barrier to entry
- Authorize access/modification of profile data
- Examples:

twitter

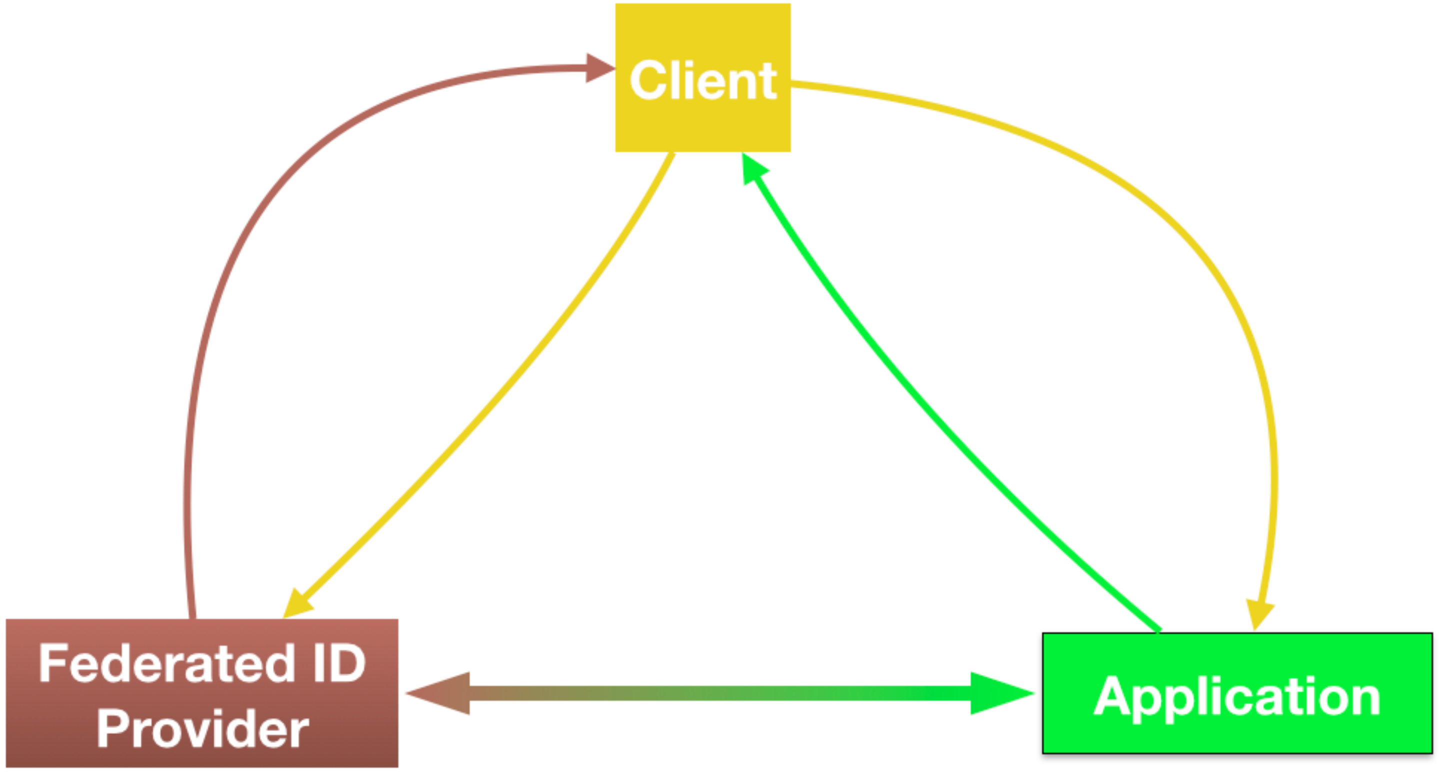
facebook

PayPal™

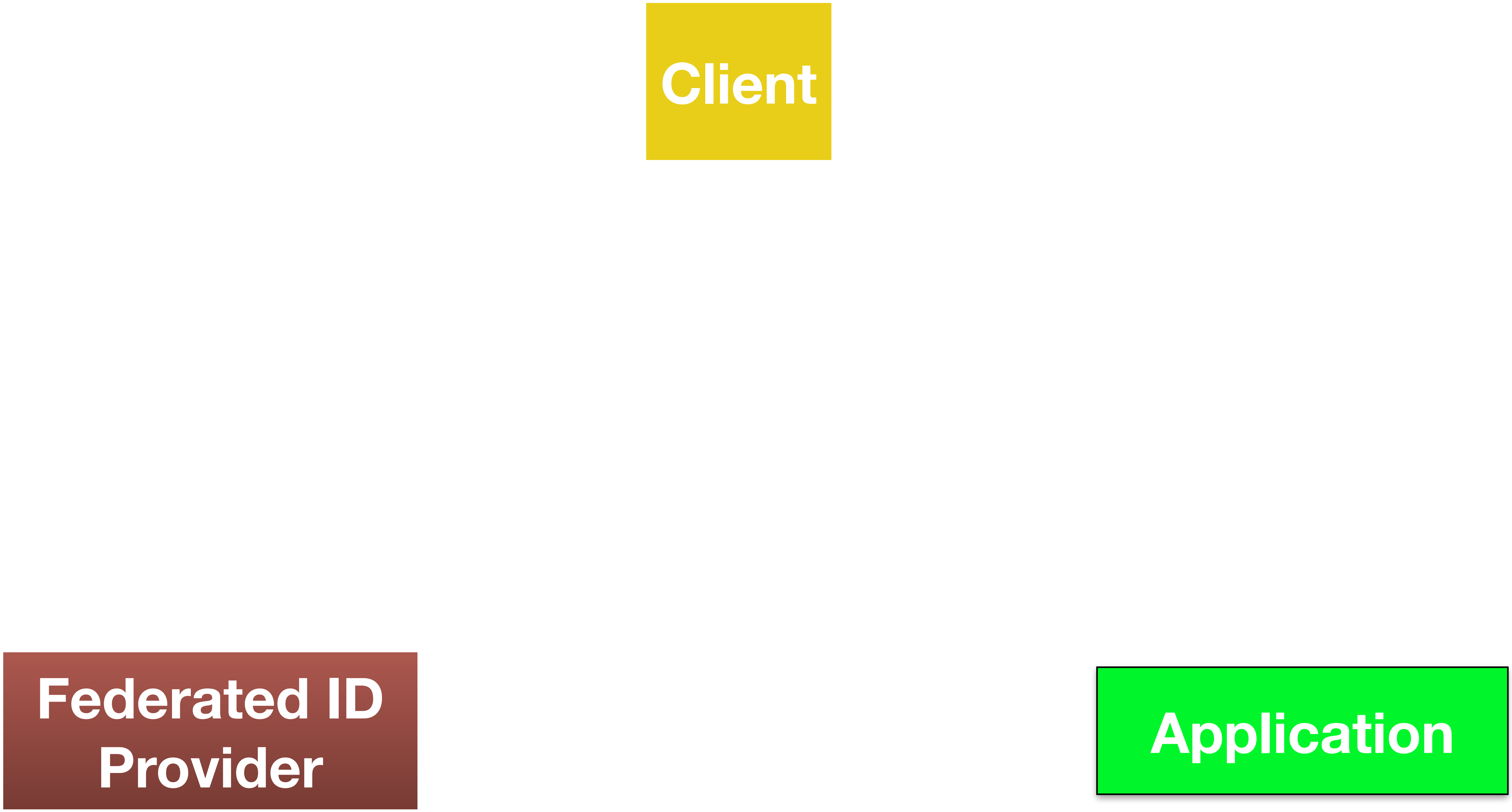
g+

Federated Authentication Interaction

High level



Federated Authentication Interaction **(high level)**

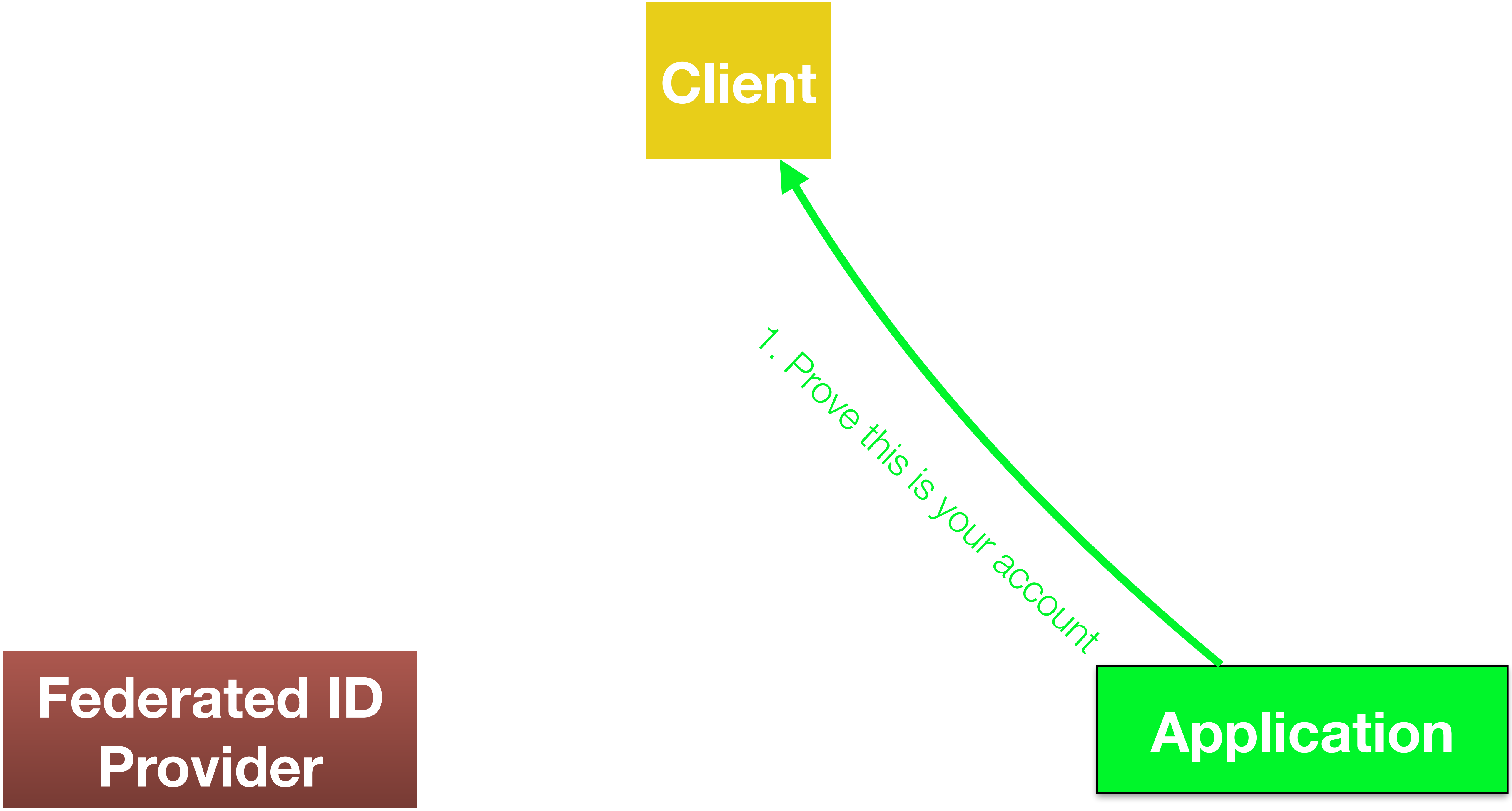


Client

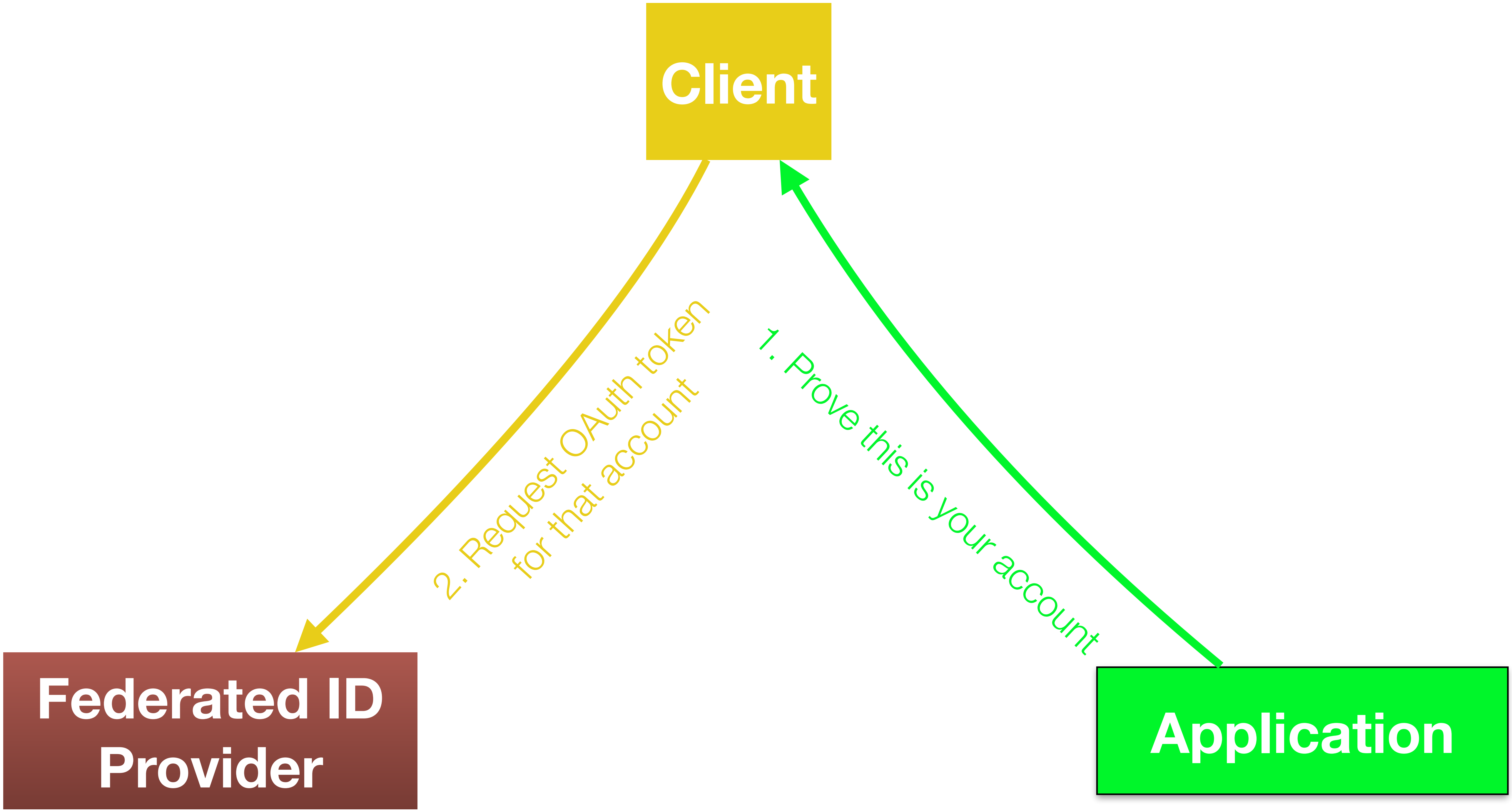
Federated ID Provider

Application

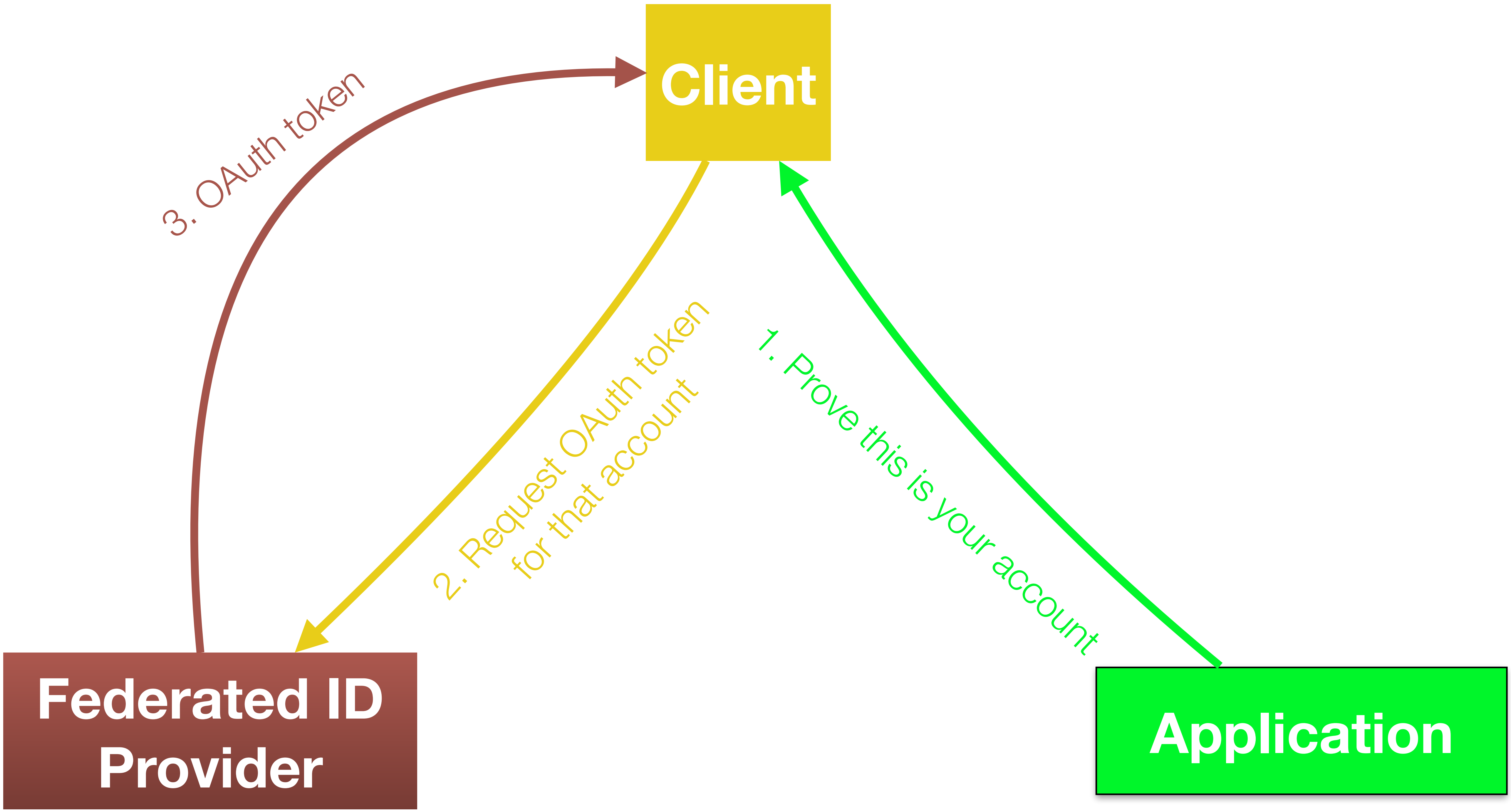
Federated Authentication Interaction **(high level)**



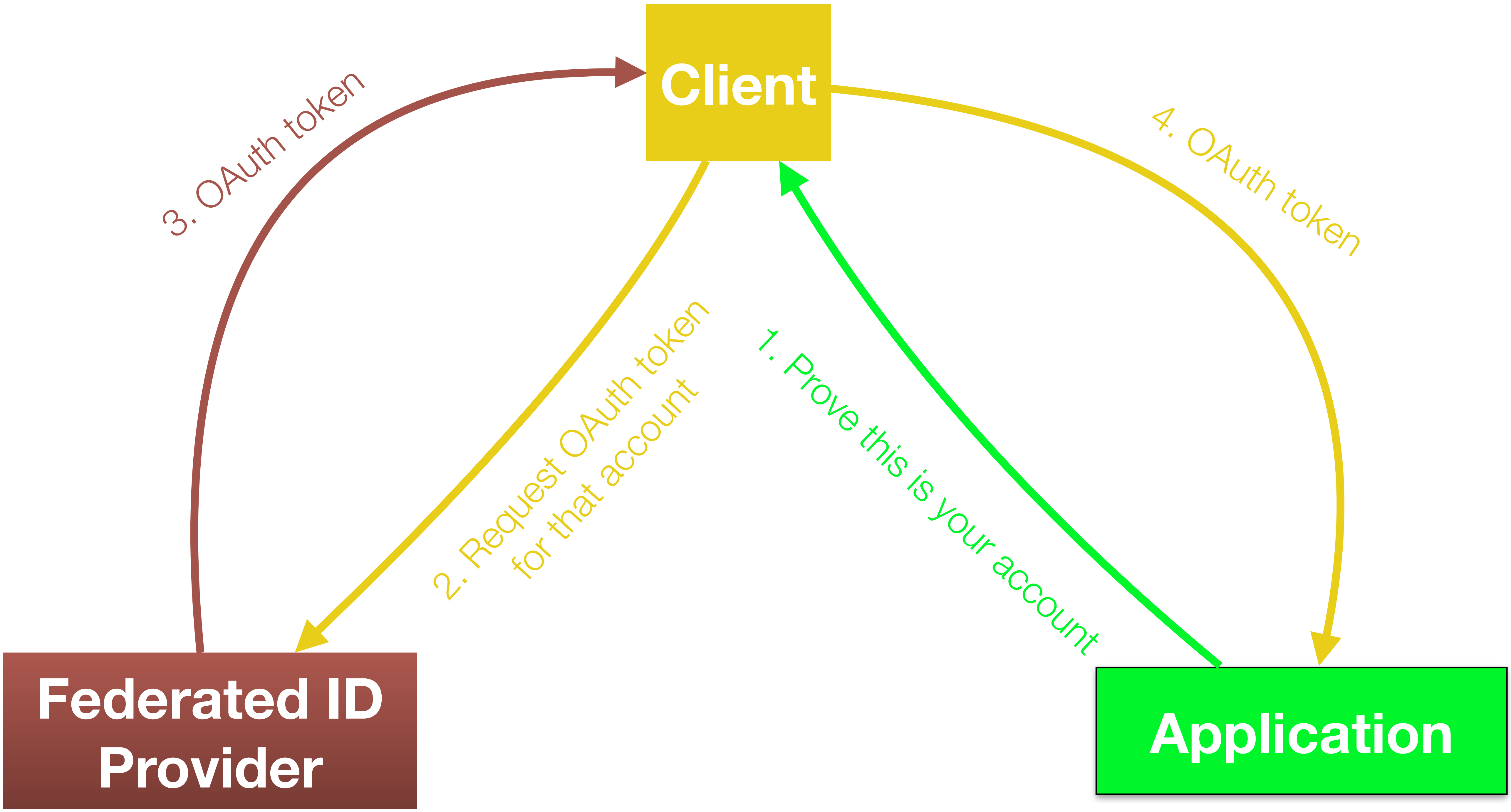
Federated Authentication Interaction **(high level)**


Client
2. Request OAuth token for that account
1. Prove this is your account
Federated ID Provider
Application

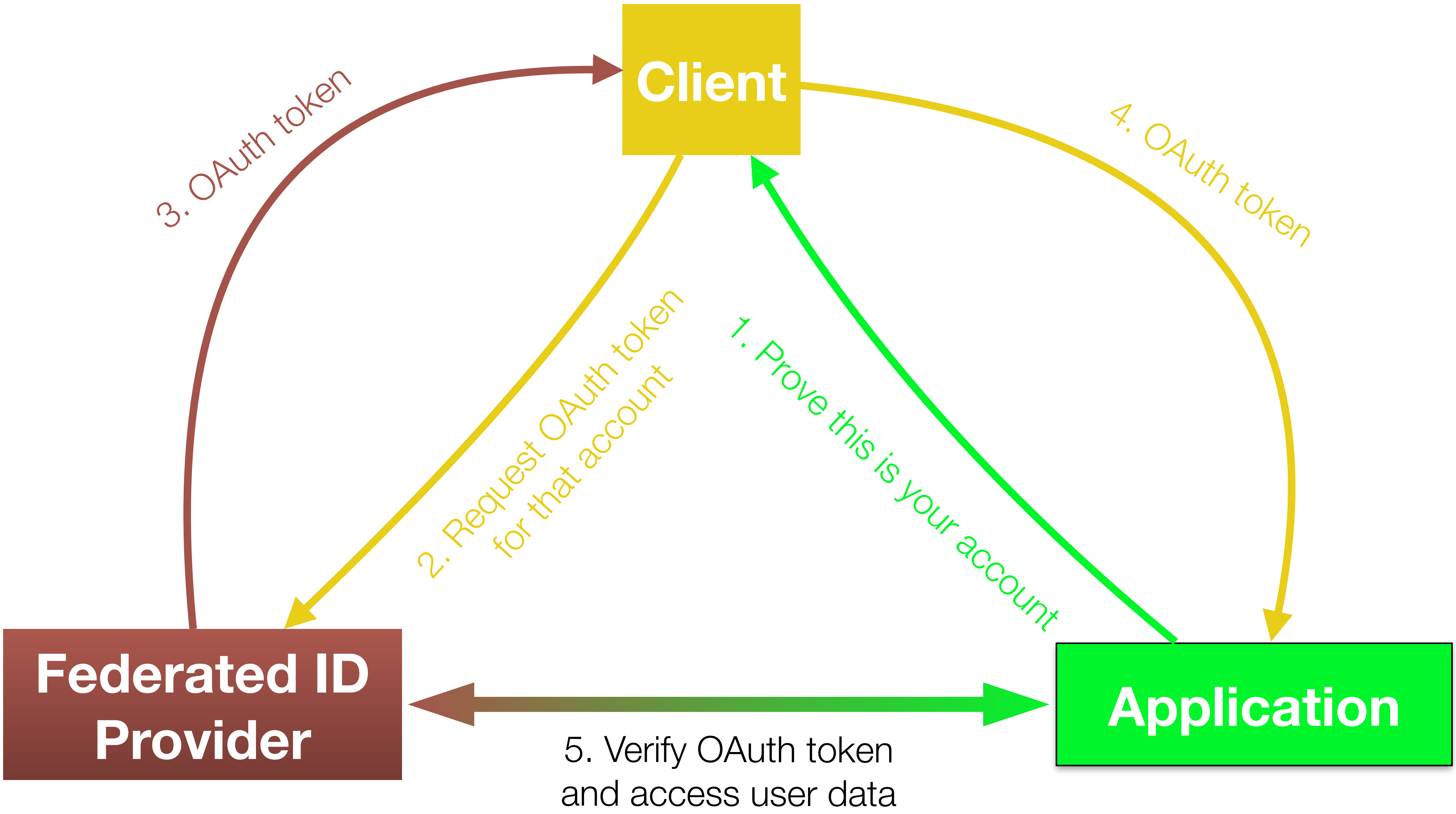
Federated Authentication Interaction **(high level)**



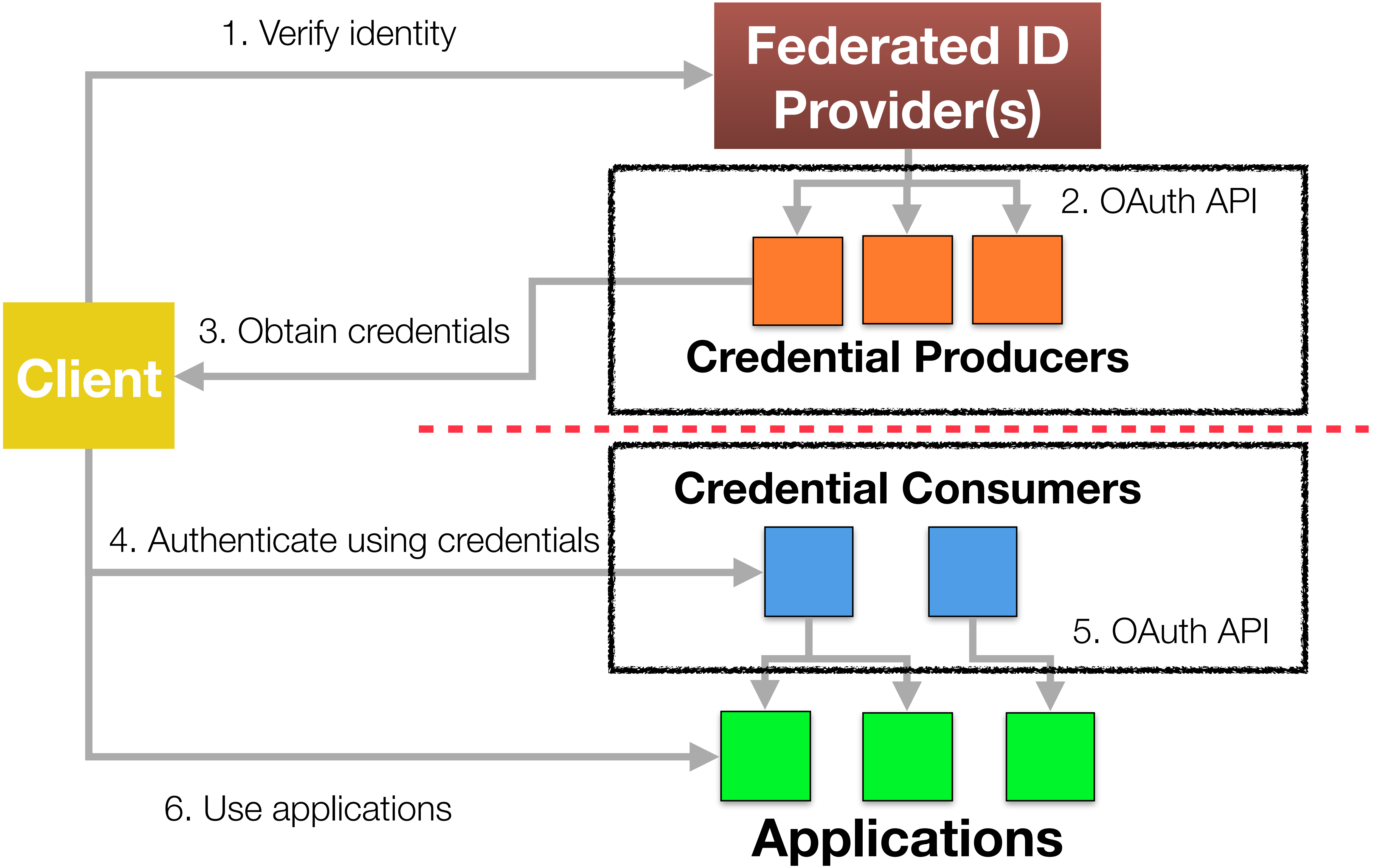
Federated Authentication Interaction **(high level)**


Client
3. OAuth token
4. OAuth token
2. Request OAuth token for that account
1. Prove this is your account
Federated ID Provider
Application

Federated Authentication Interaction **(high level)**

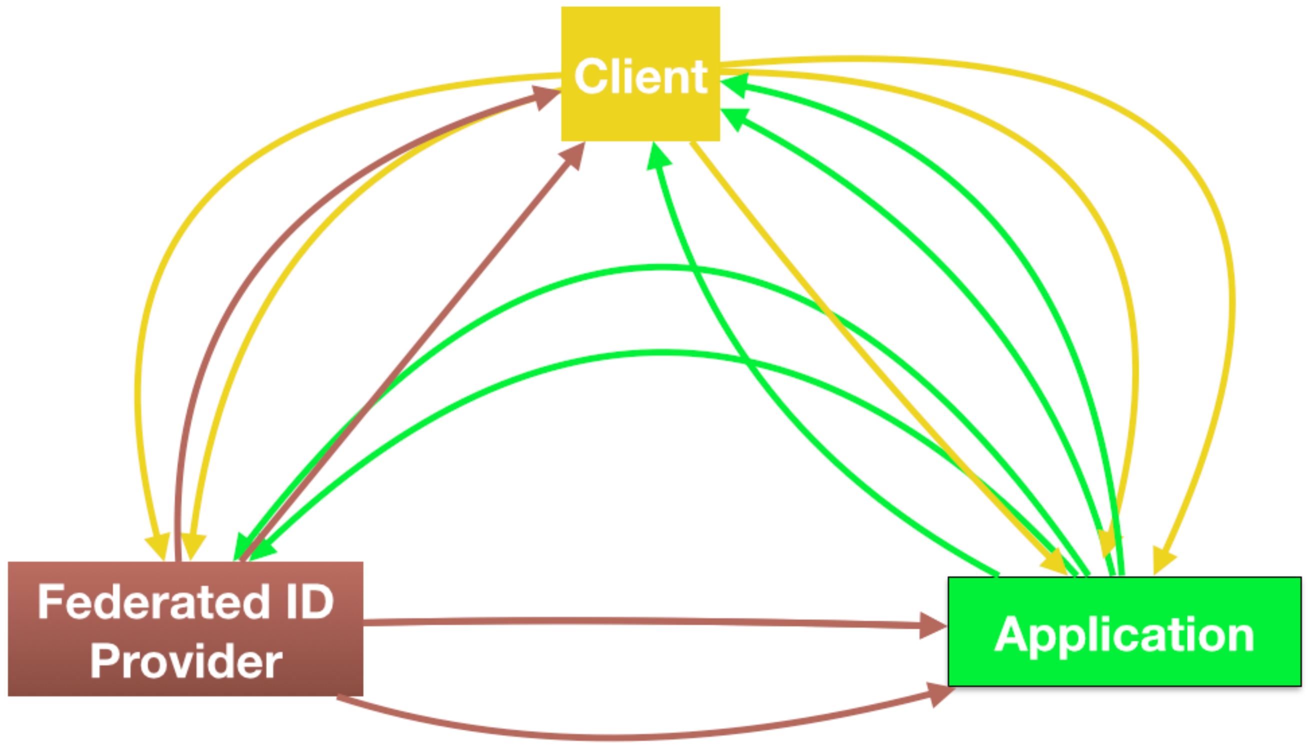

Client
3. OAuth token
4. OAuth token
2. Request OAuth token for that account
1. Prove this is your account
Federated ID Provider
Application
5. Verify OAuth token and access user data

System Architecture

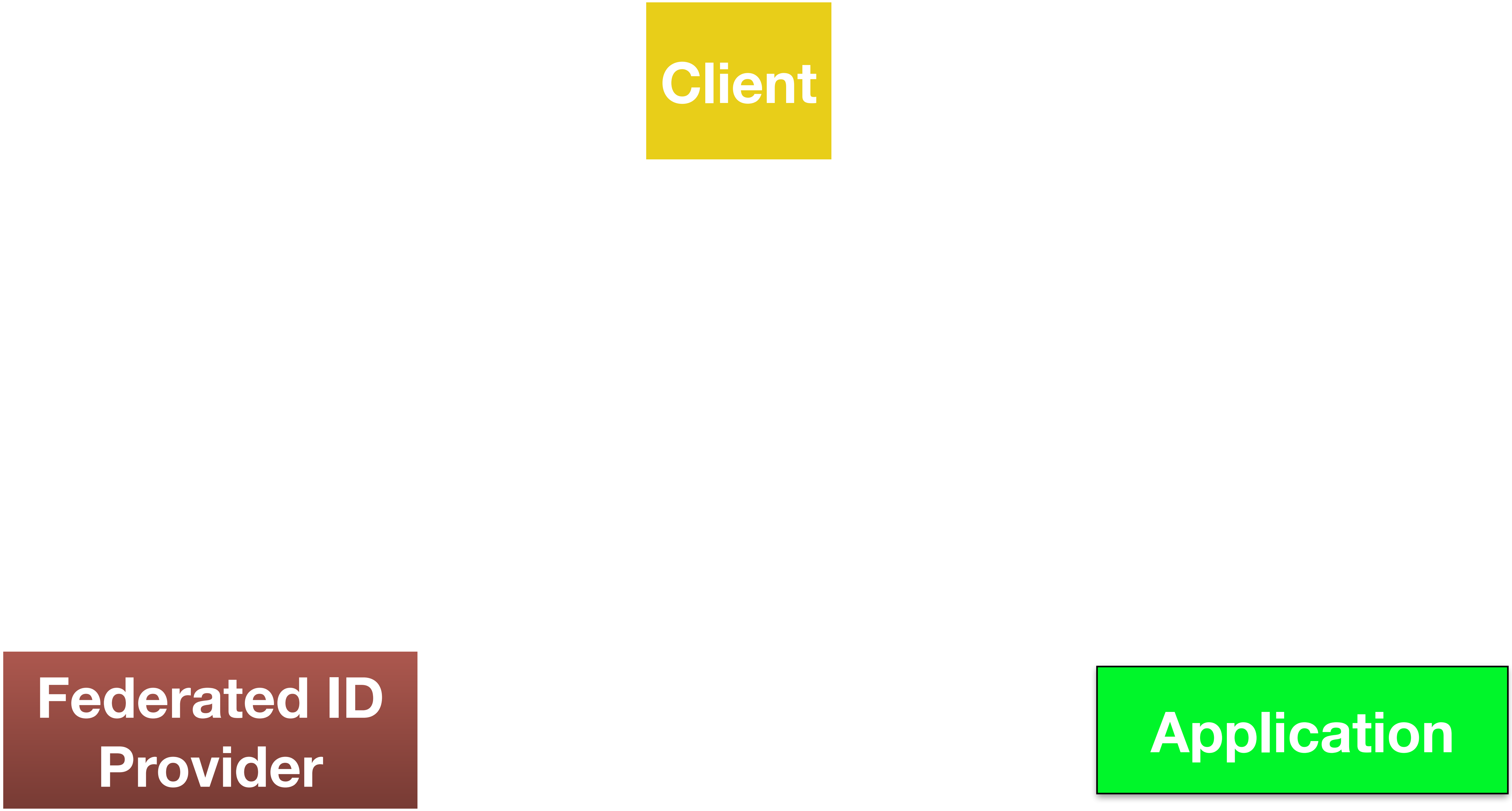


Federated ID Authentication

Detailed view



Federated Authentication Interaction

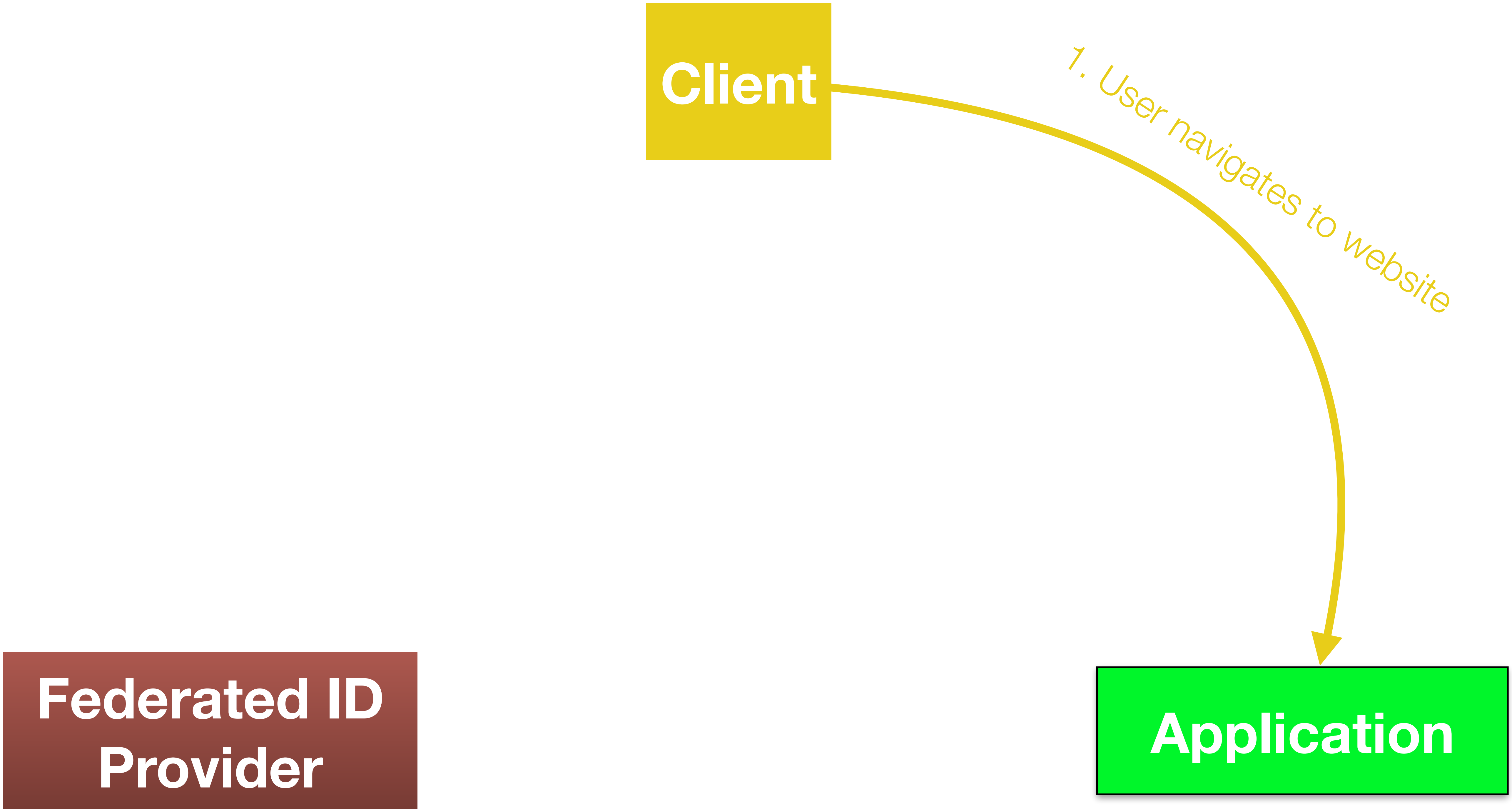


Client

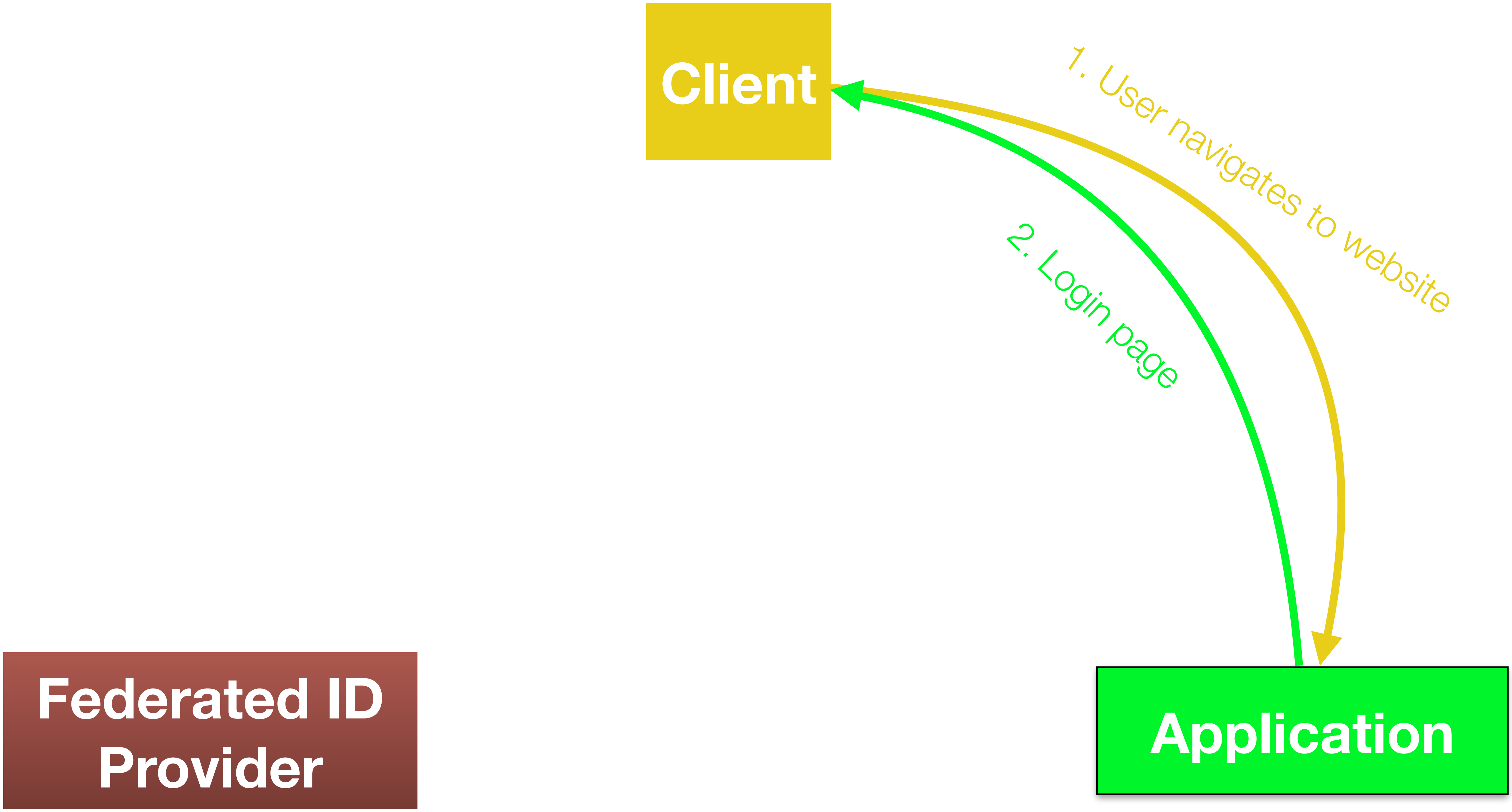
Federated ID Provider

Application

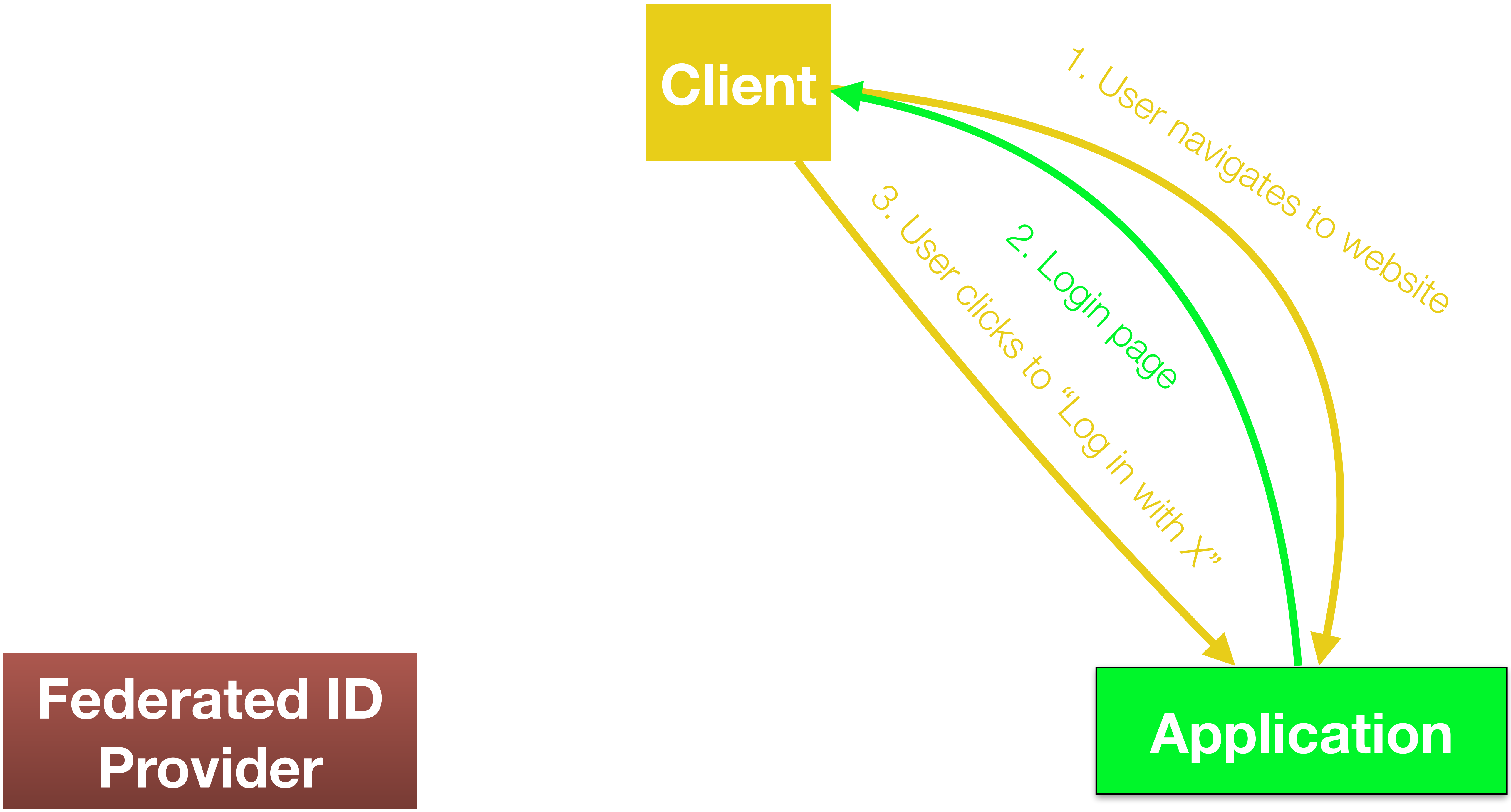
Federated Authentication Interaction



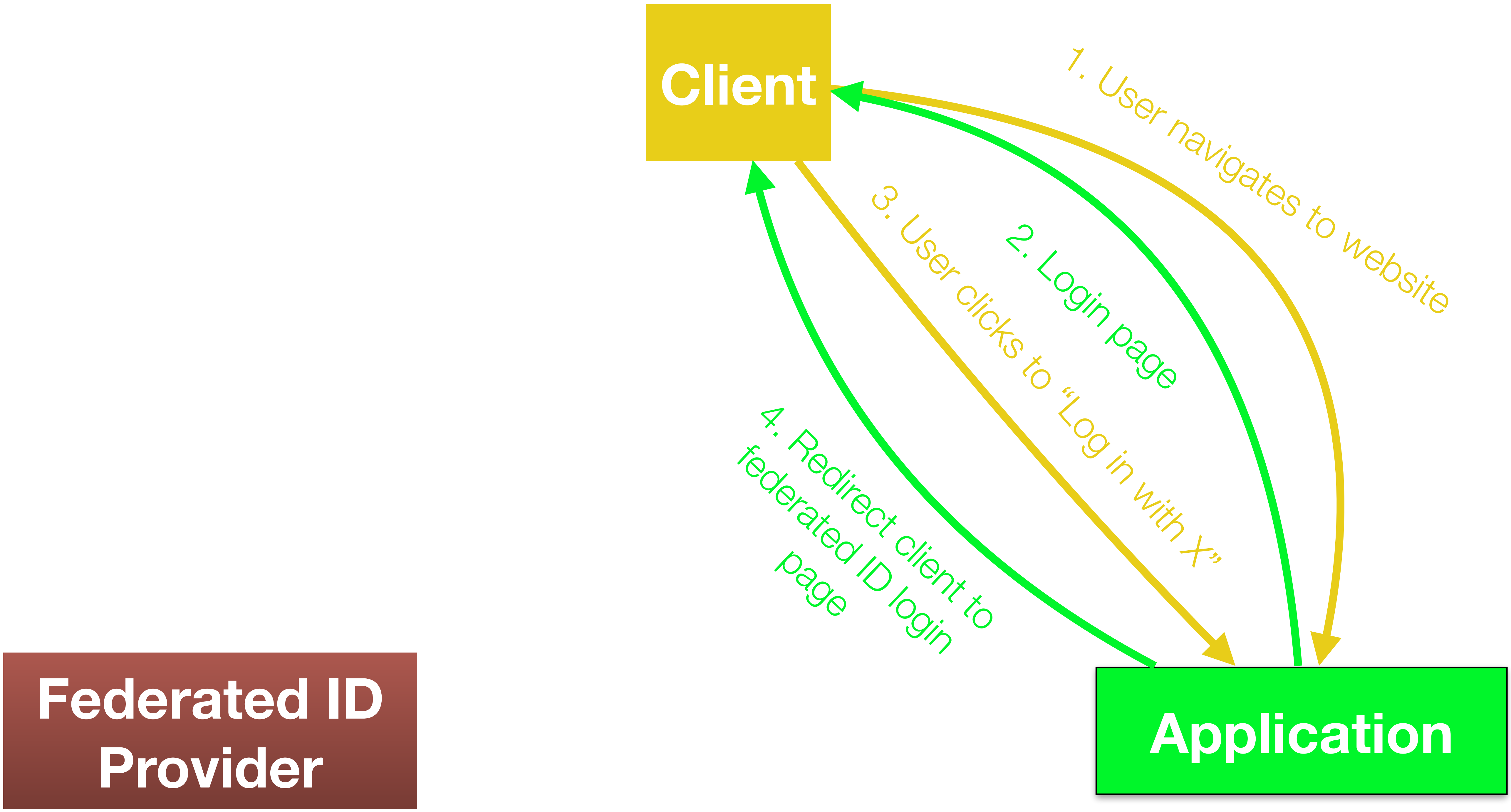
Federated Authentication Interaction



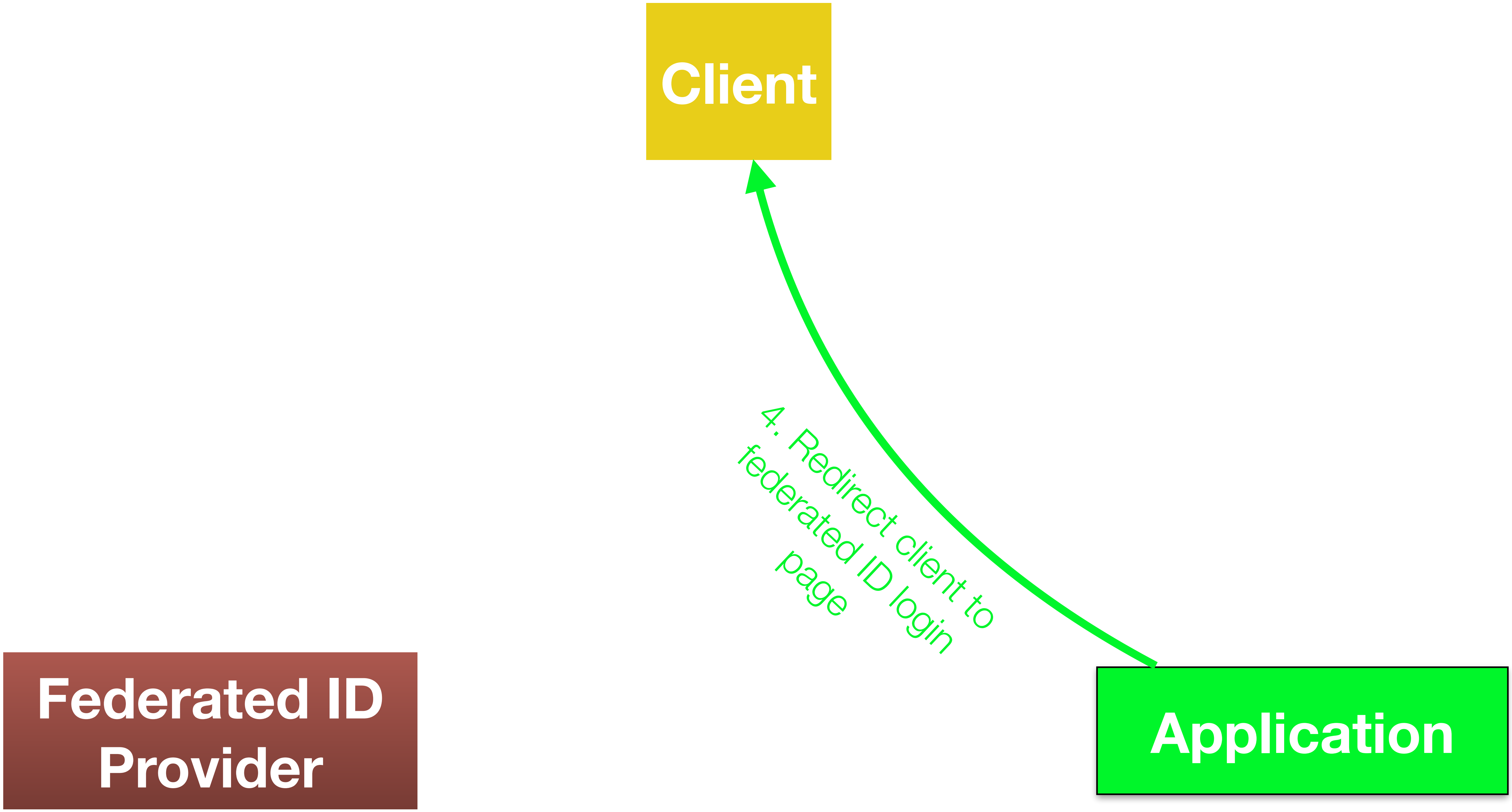
Federated Authentication Interaction


Client
1. User navigates to website
2. Login page
3. User clicks to "Log in with X"
Federated ID Provider
Application

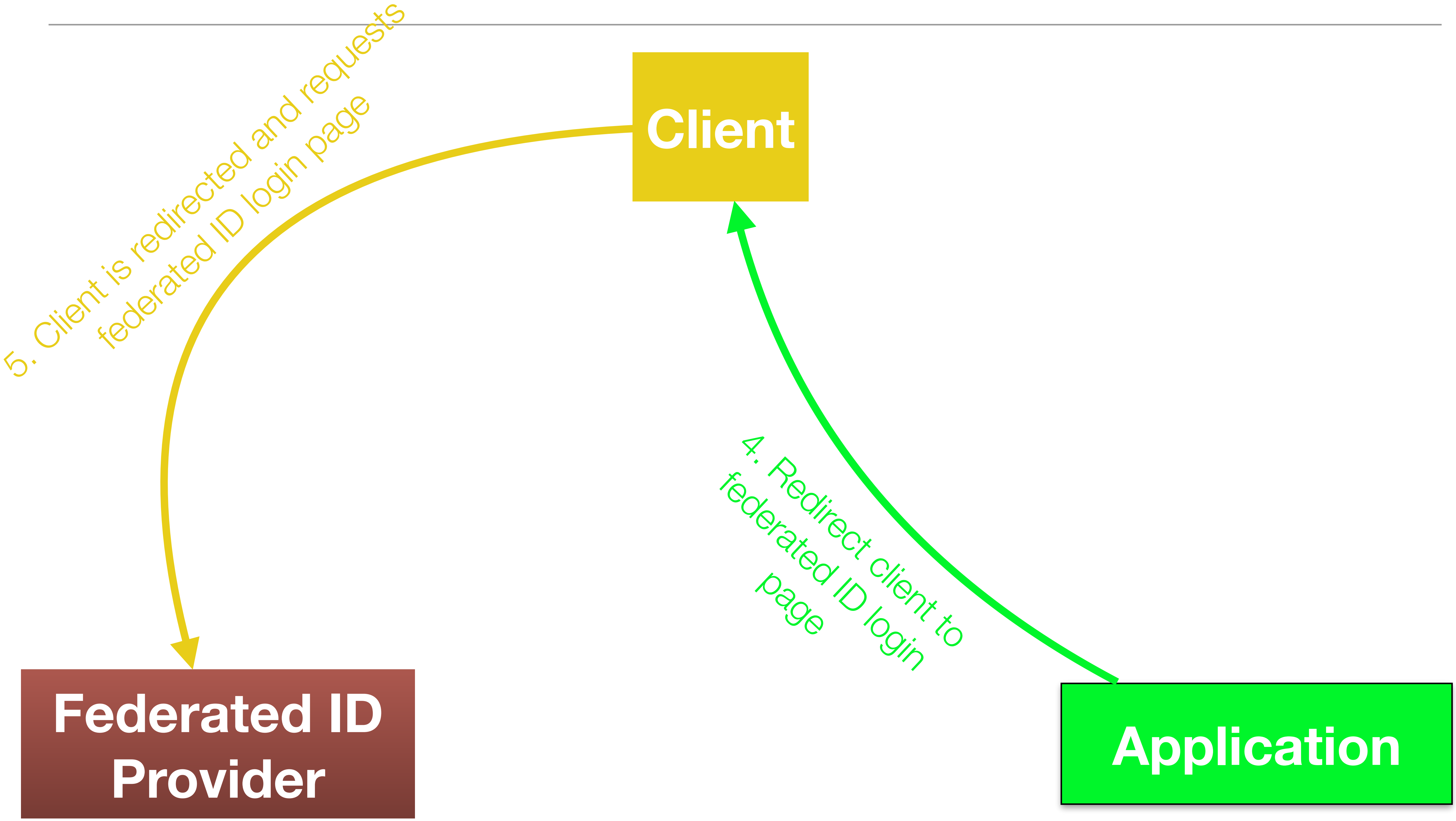
Federated Authentication Interaction


Client
1. User navigates to website
2. Login page
3. User clicks to "Log in with X"
4. Redirect client to federated ID login page
Federated ID Provider
Application

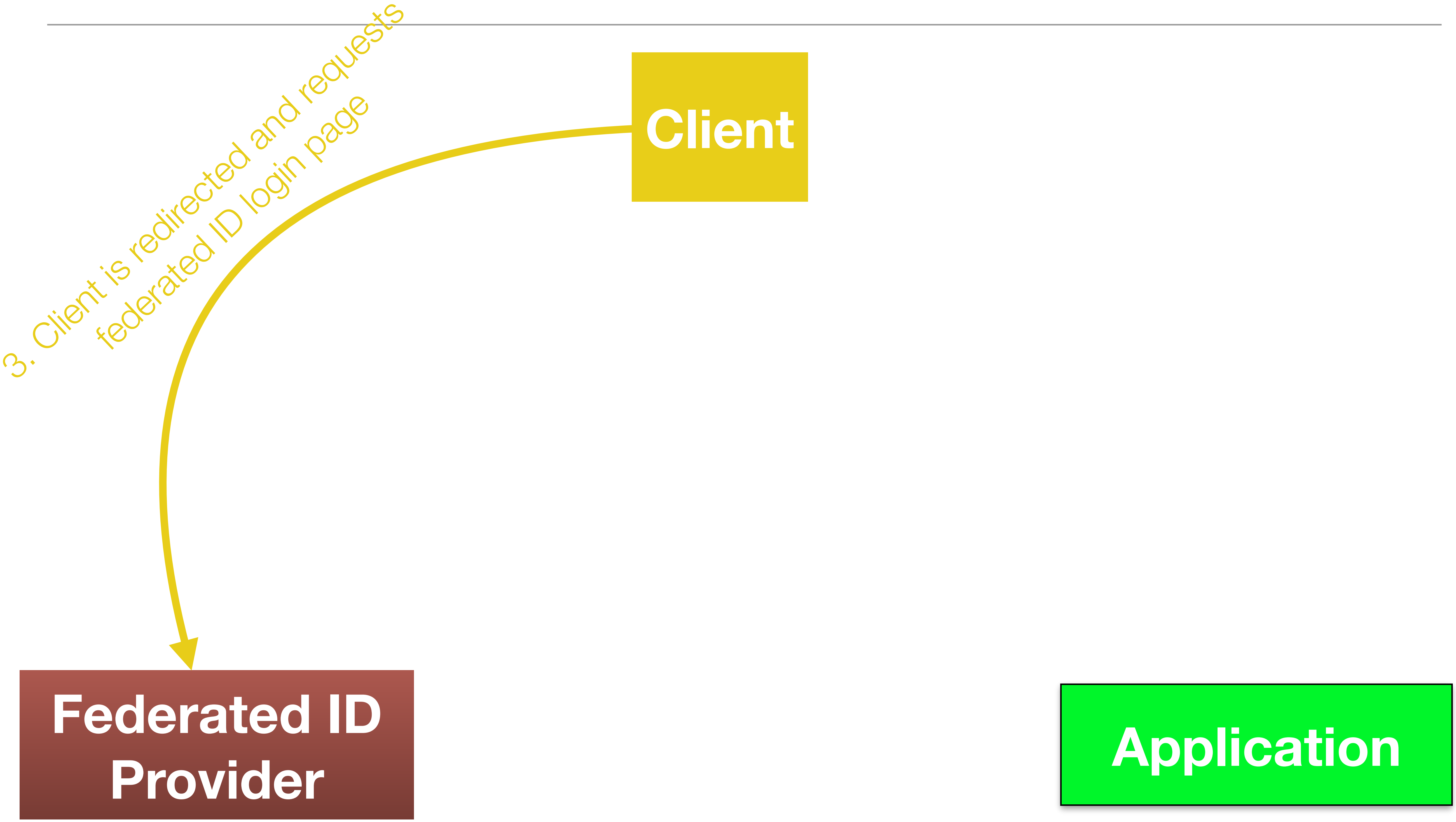
Federated Authentication Interaction



Federated Authentication Interaction

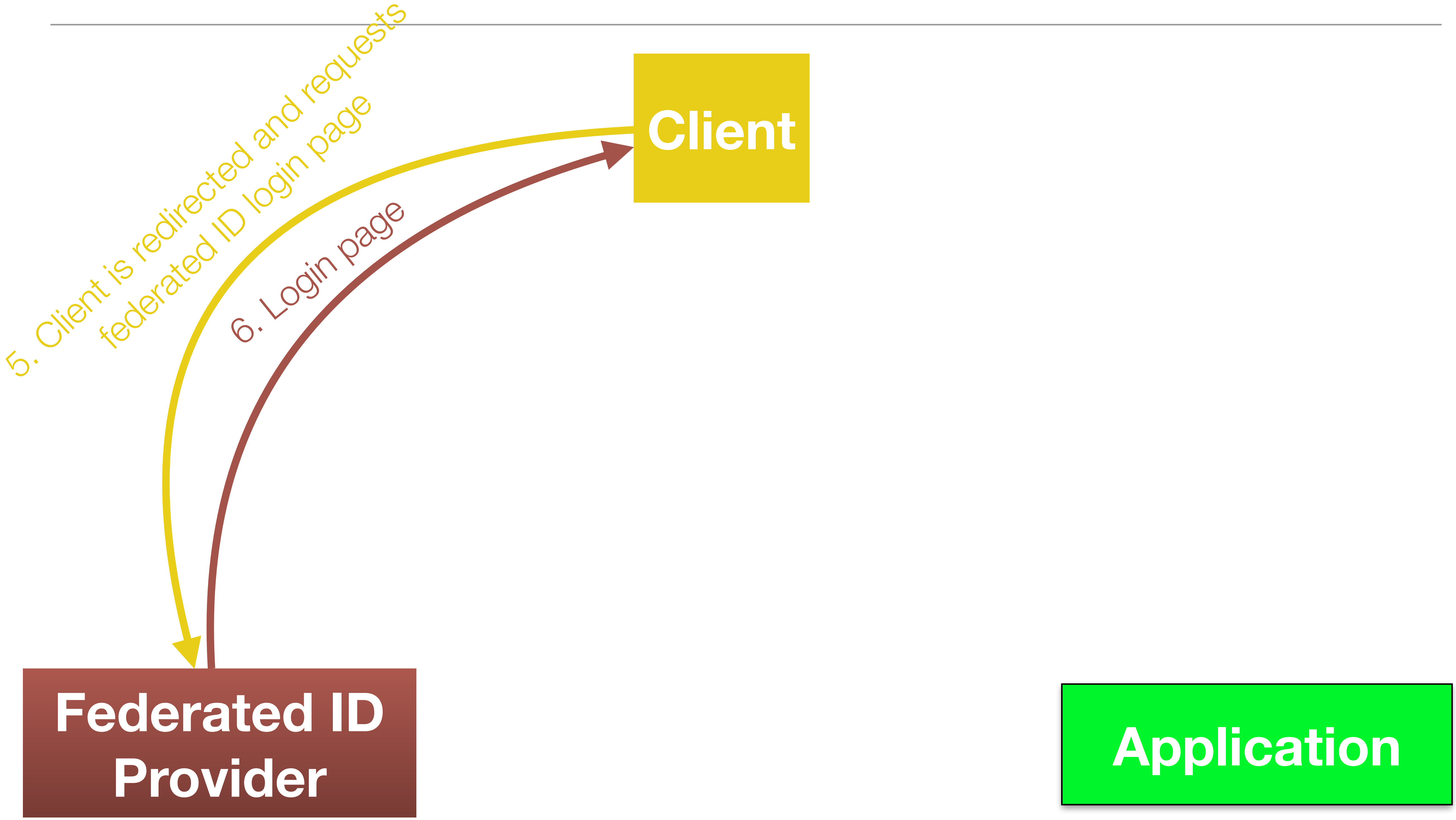

Client
5. Client is redirected and requests federated ID login page
4. Redirect client to federated ID login page
Federated ID Provider
Application

Federated Authentication Interaction



Client
3. Client is redirected and requests federated ID login page
Federated ID Provider
Application

Federated Authentication Interaction



Client

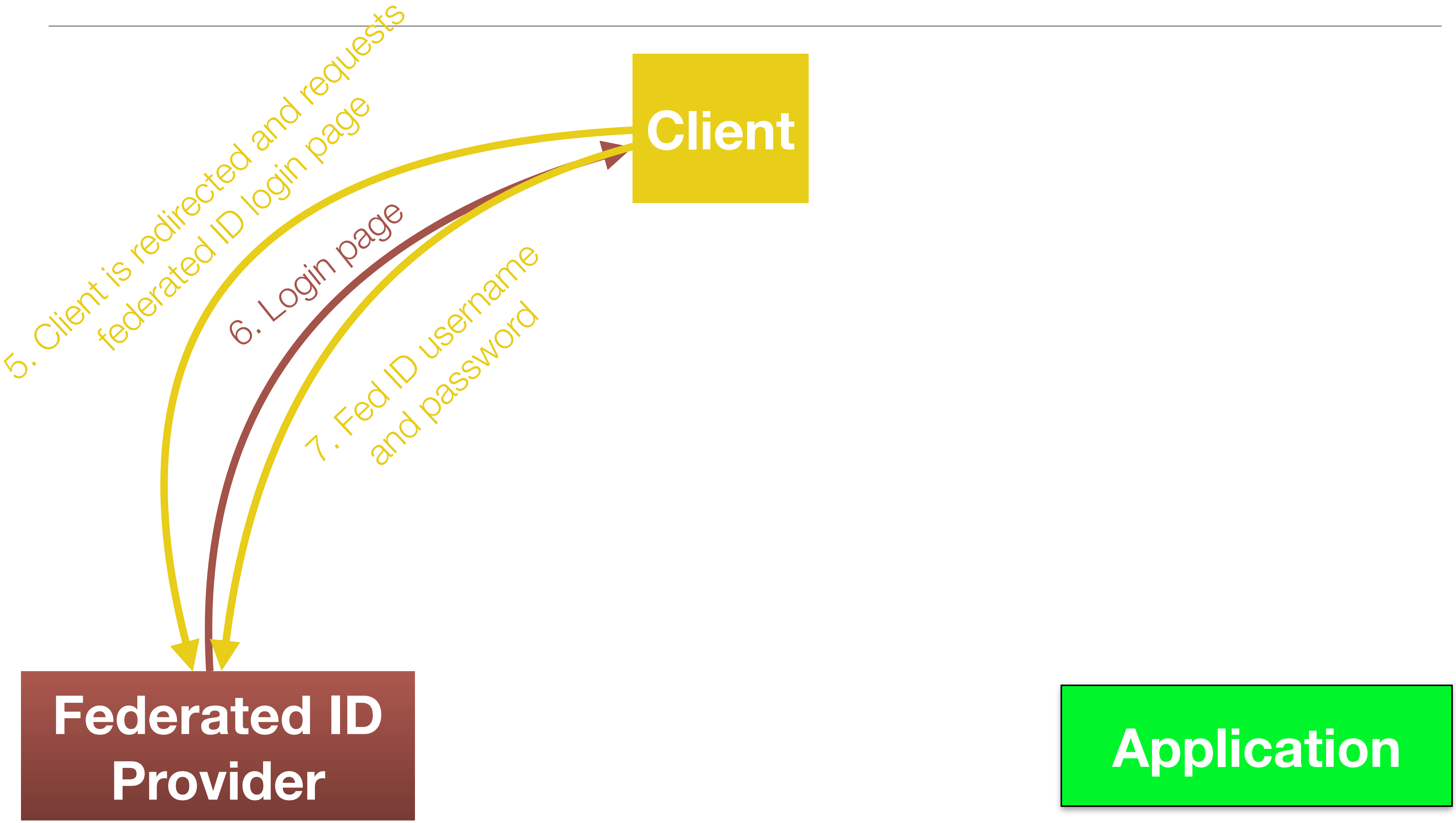
5. Client is redirected and requests federated ID login page

6. Login page

Federated ID Provider

Application

Federated Authentication Interaction



Client

5. Client is redirected and requests federated ID login page

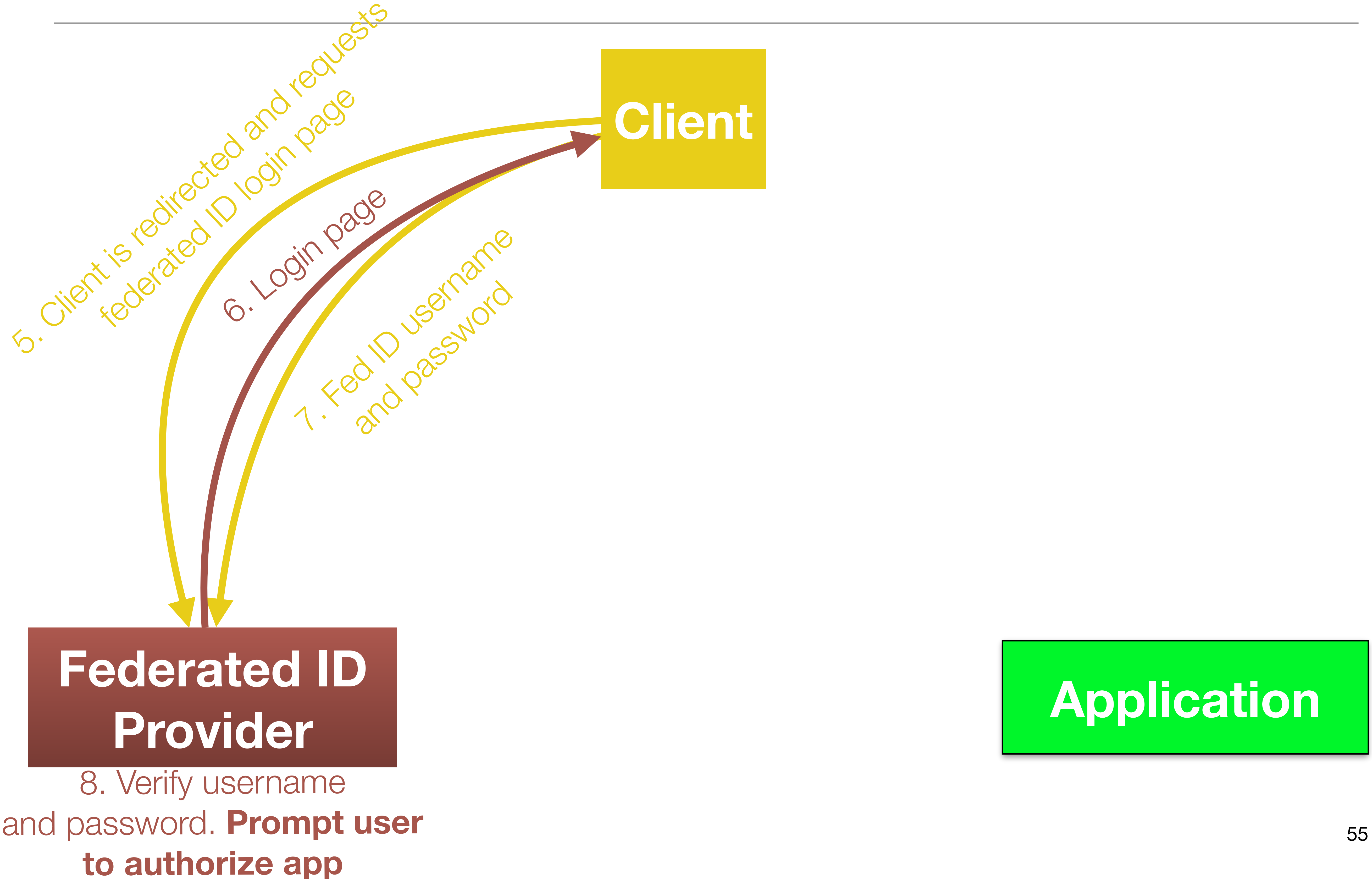
6. Login page

7. Fed ID username and password

Federated ID Provider

Application

Federated Authentication Interaction



Client

5. Client is redirected and requests federated ID login page

6. Login page

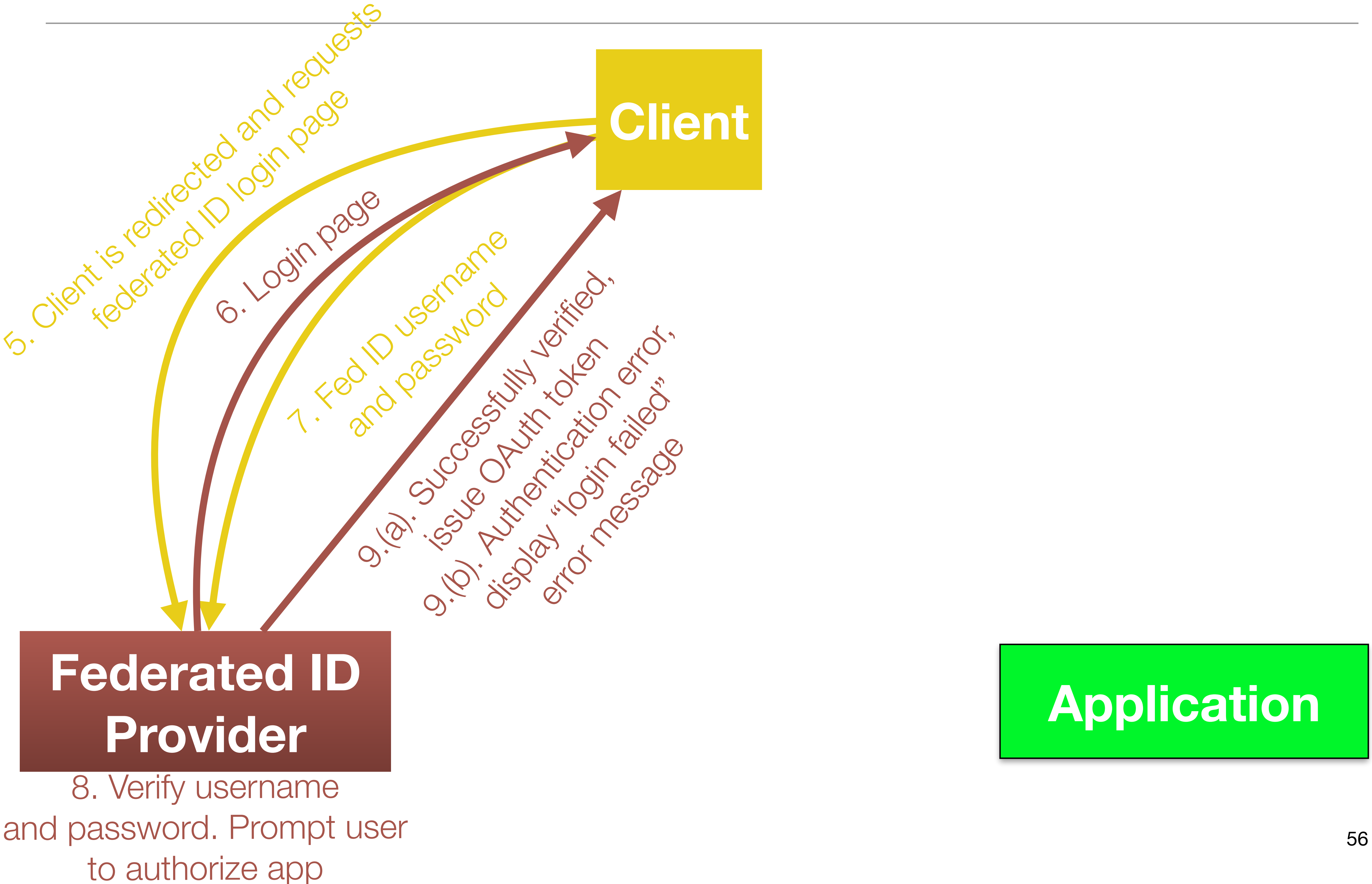
7. Fed ID username and password

Federated ID Provider

8. Verify username and password. **Prompt user to authorize app**

Application

Federated Authentication Interaction



Client

Federated ID Provider

Application

5. Client is redirected and requests federated ID login page

6. Login page

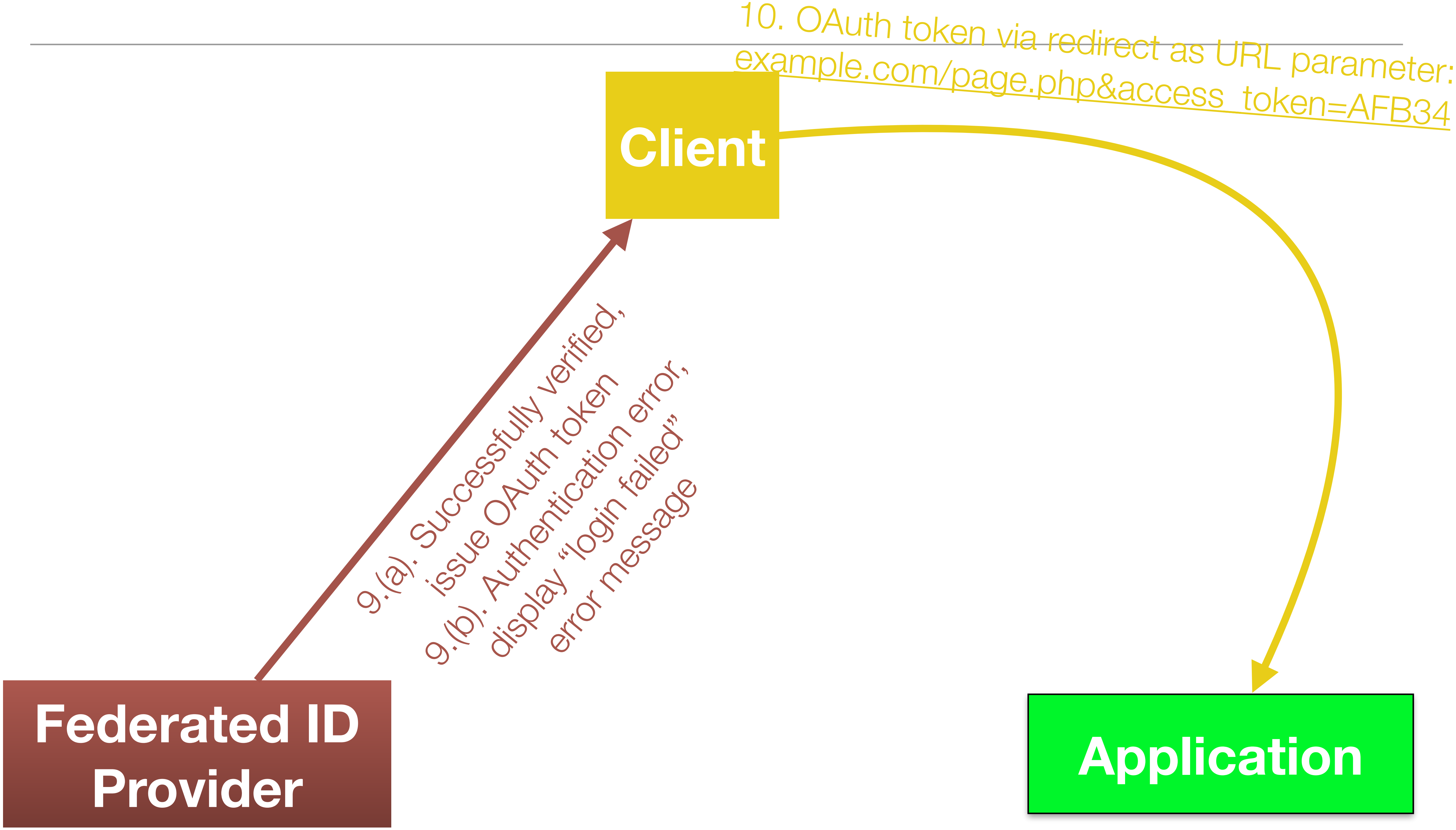
7. Fed ID username and password

8. Verify username and password. Prompt user to authorize app

9.(a). Successfully verified, issue OAuth token

9.(b). Authentication error, display “login failed” error message

Federated Authentication Interaction



10. OAuth token via redirect as URL parameter:
example.com/page.php&access_token=AFB34

Client

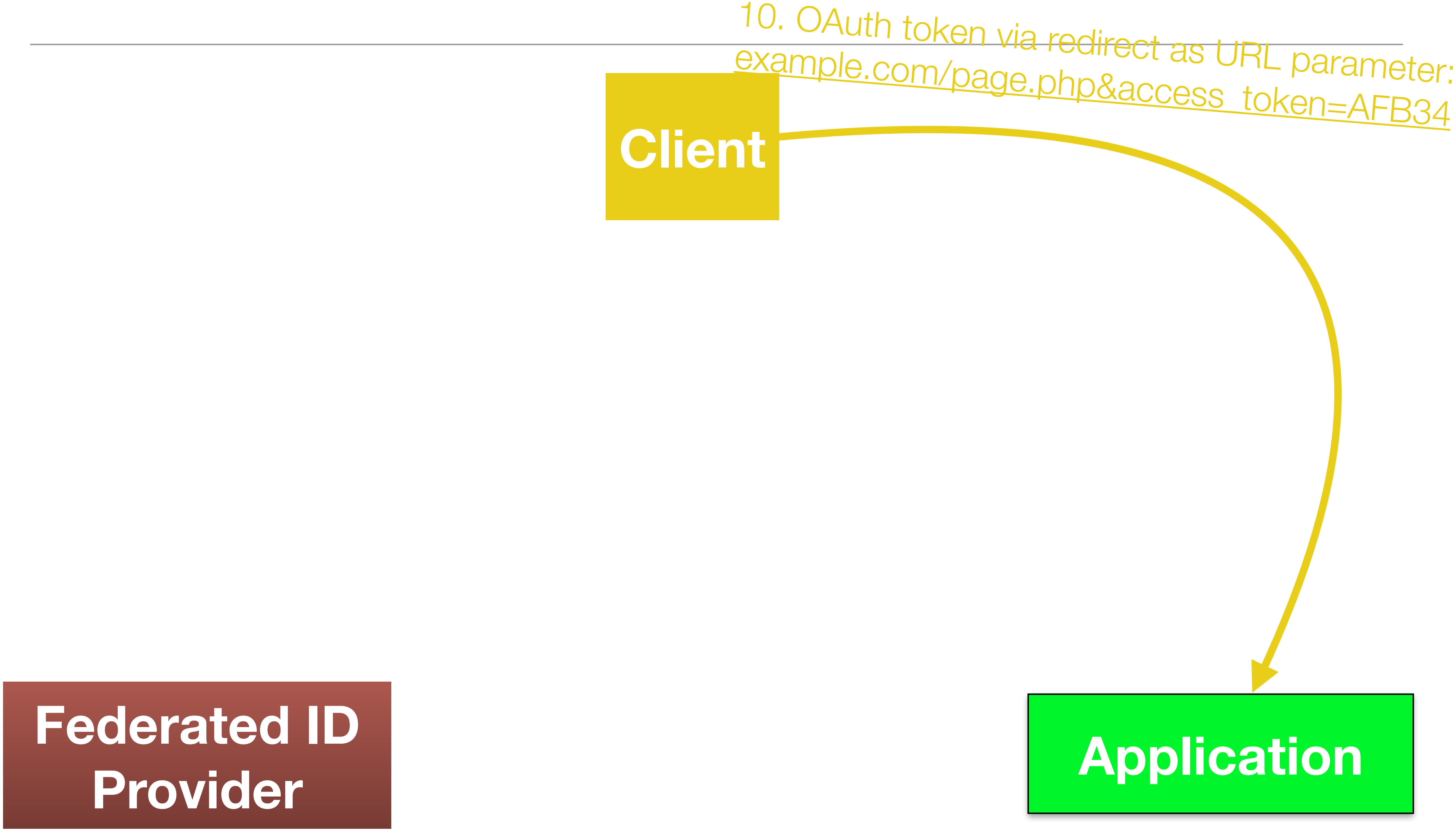
9.(a). Successfully verified, issue OAuth token

9.(b). Authentication error, display “login failed” error message

Federated ID Provider

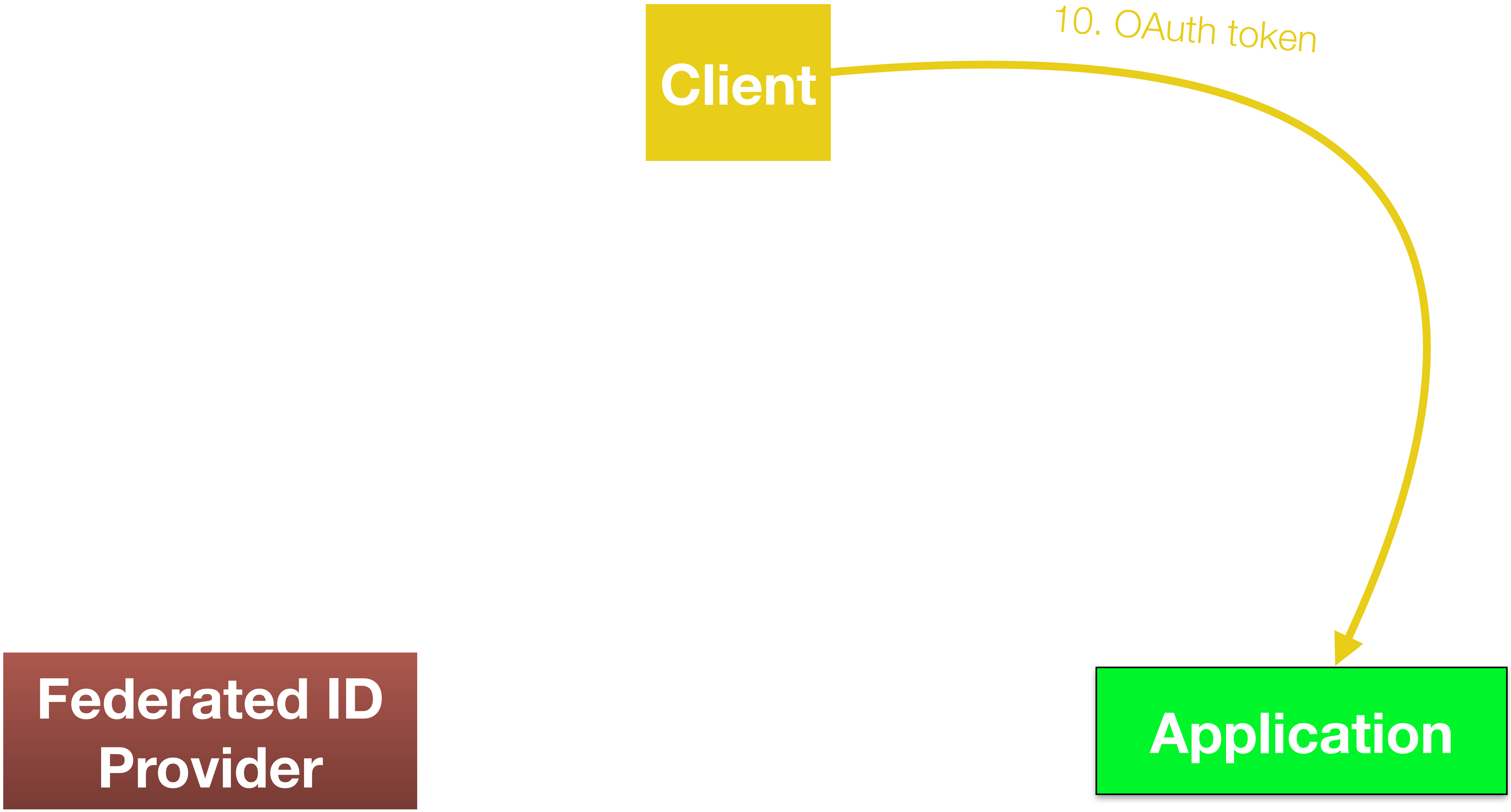
Application

Federated Authentication Interaction

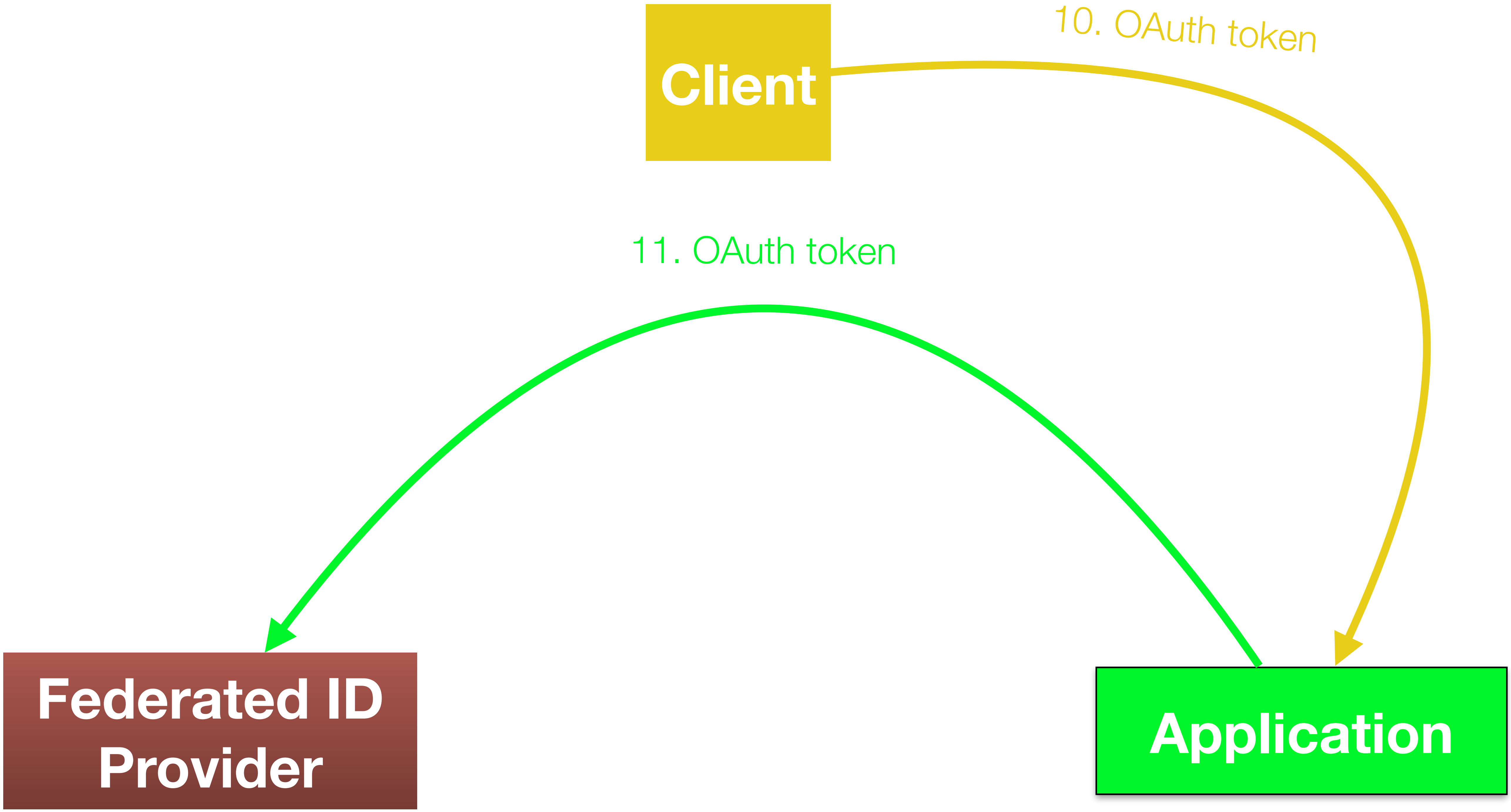


10. OAuth token via redirect as URL parameter:
example.com/page.php&access_token=AFB34

Federated Authentication Interaction



Federated Authentication Interaction


Client

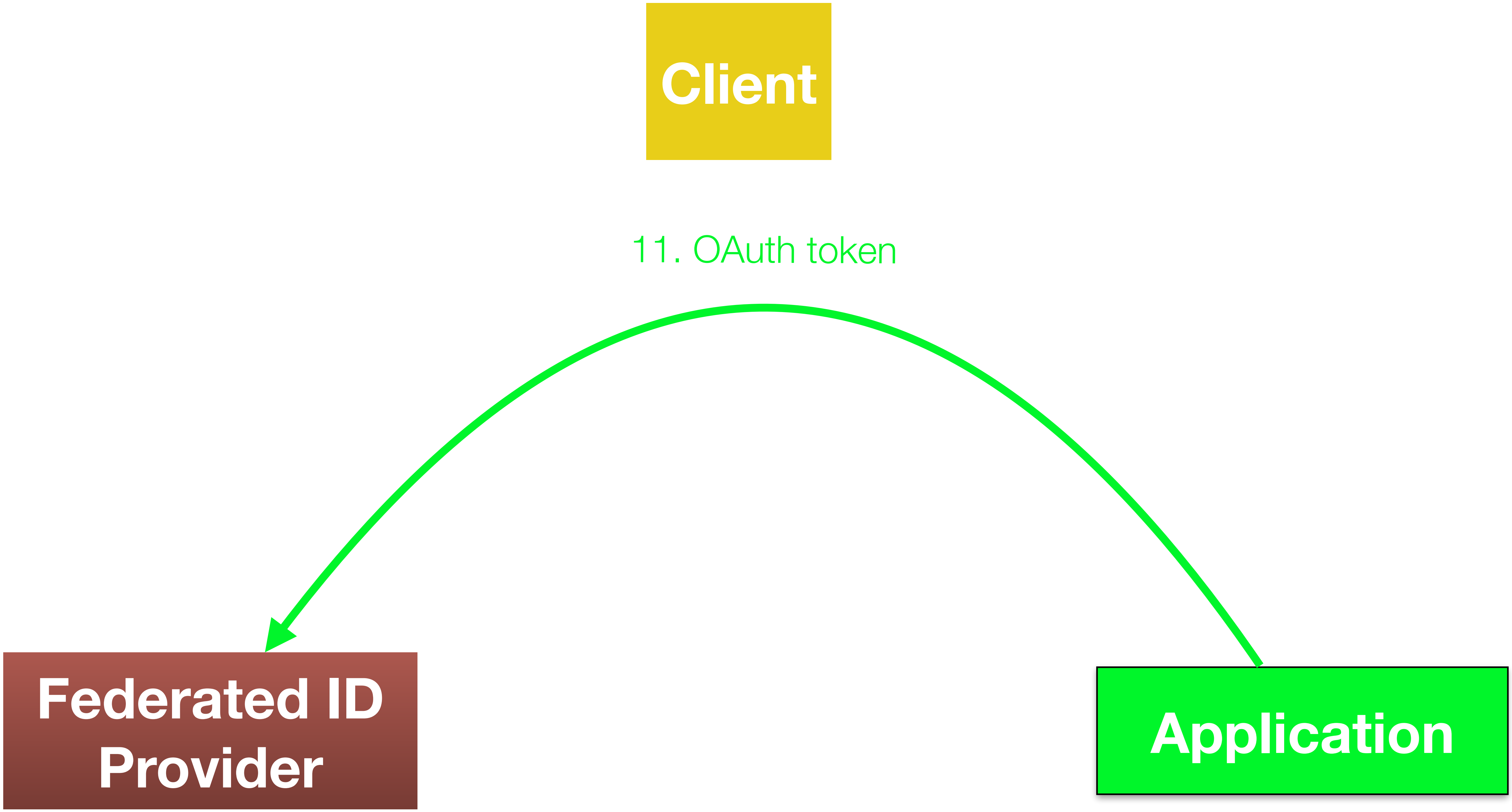
10. OAuth token

11. OAuth token

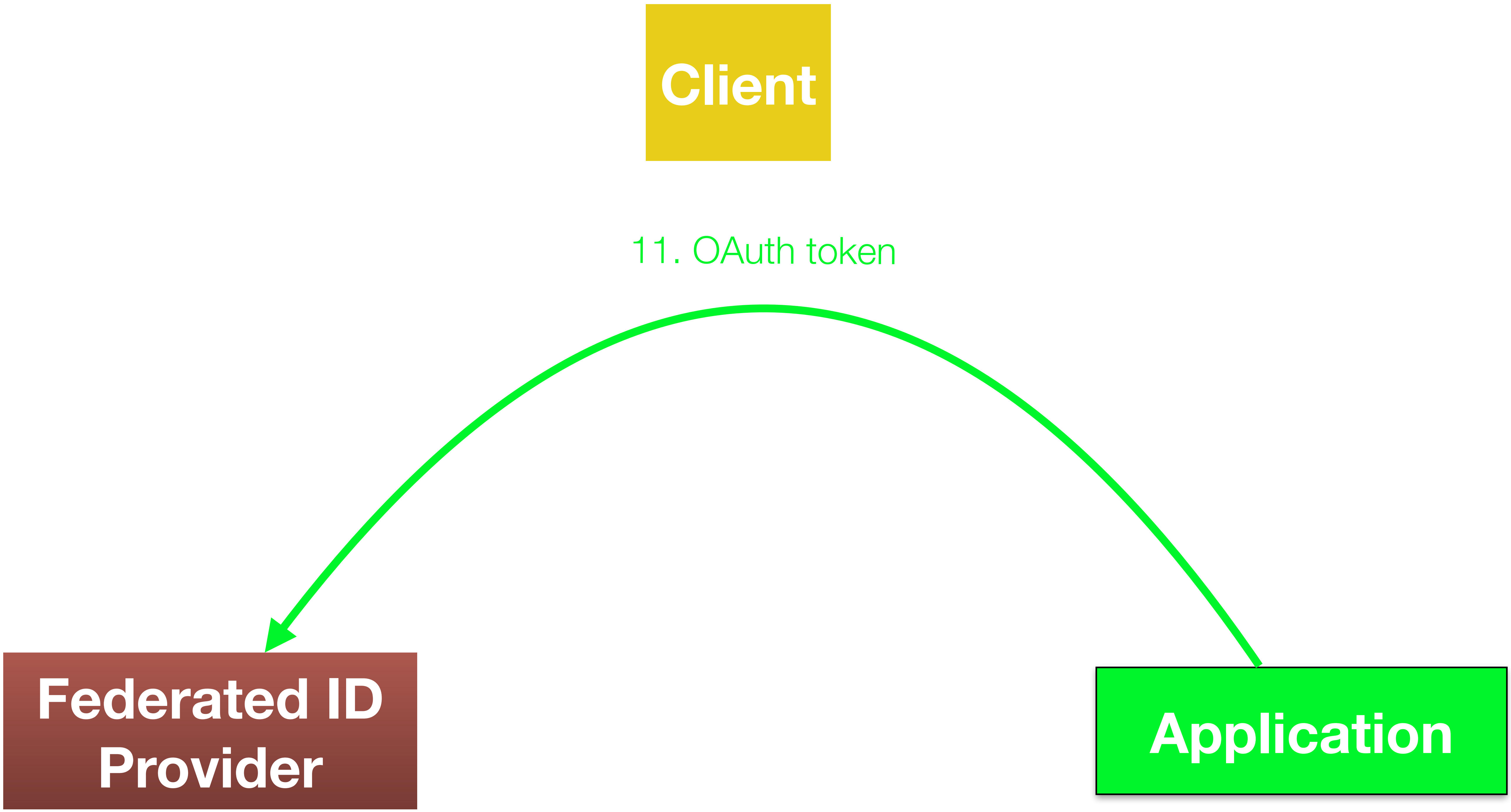
Federated ID Provider

Application

Federated Authentication Interaction

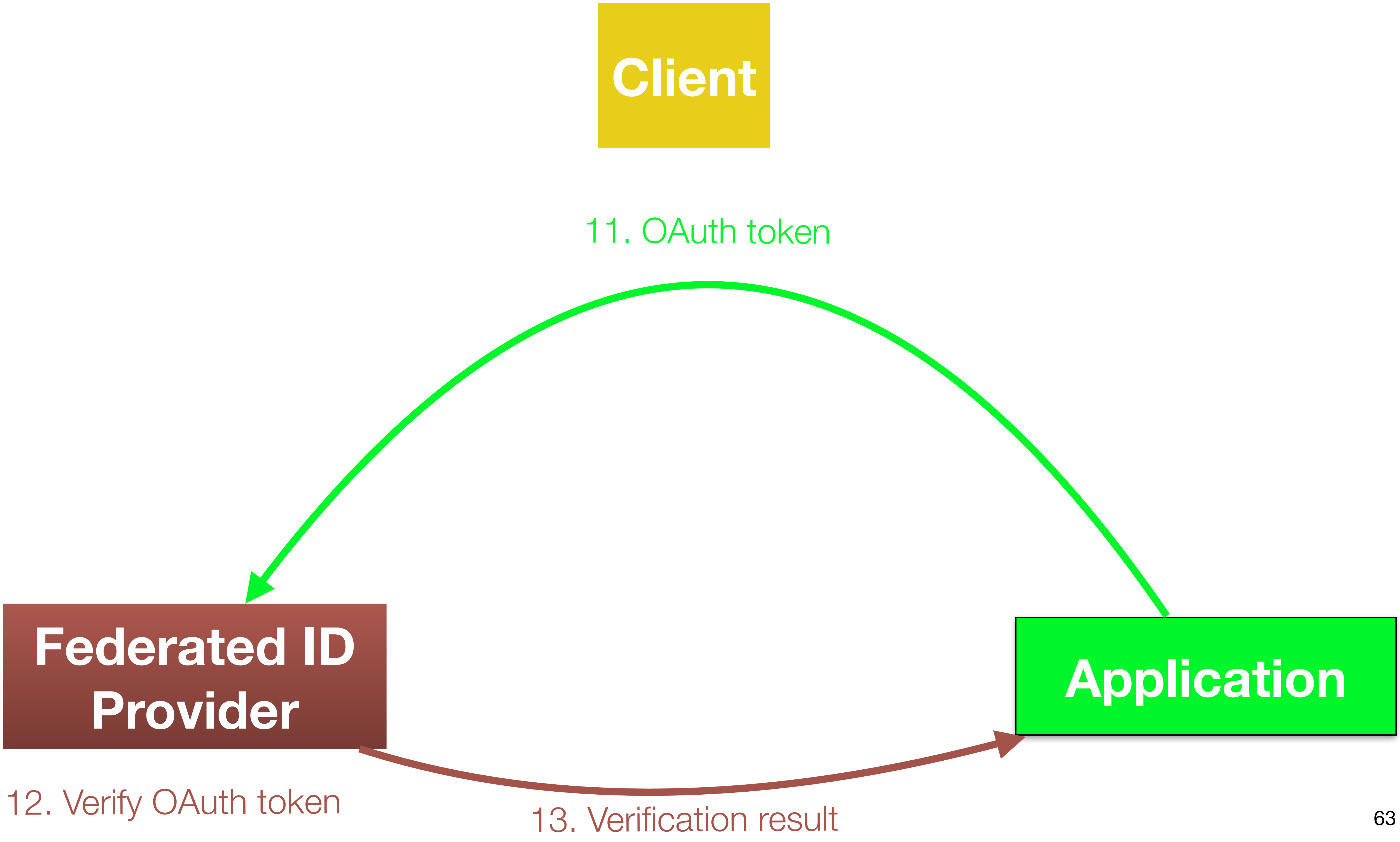


Federated Authentication Interaction

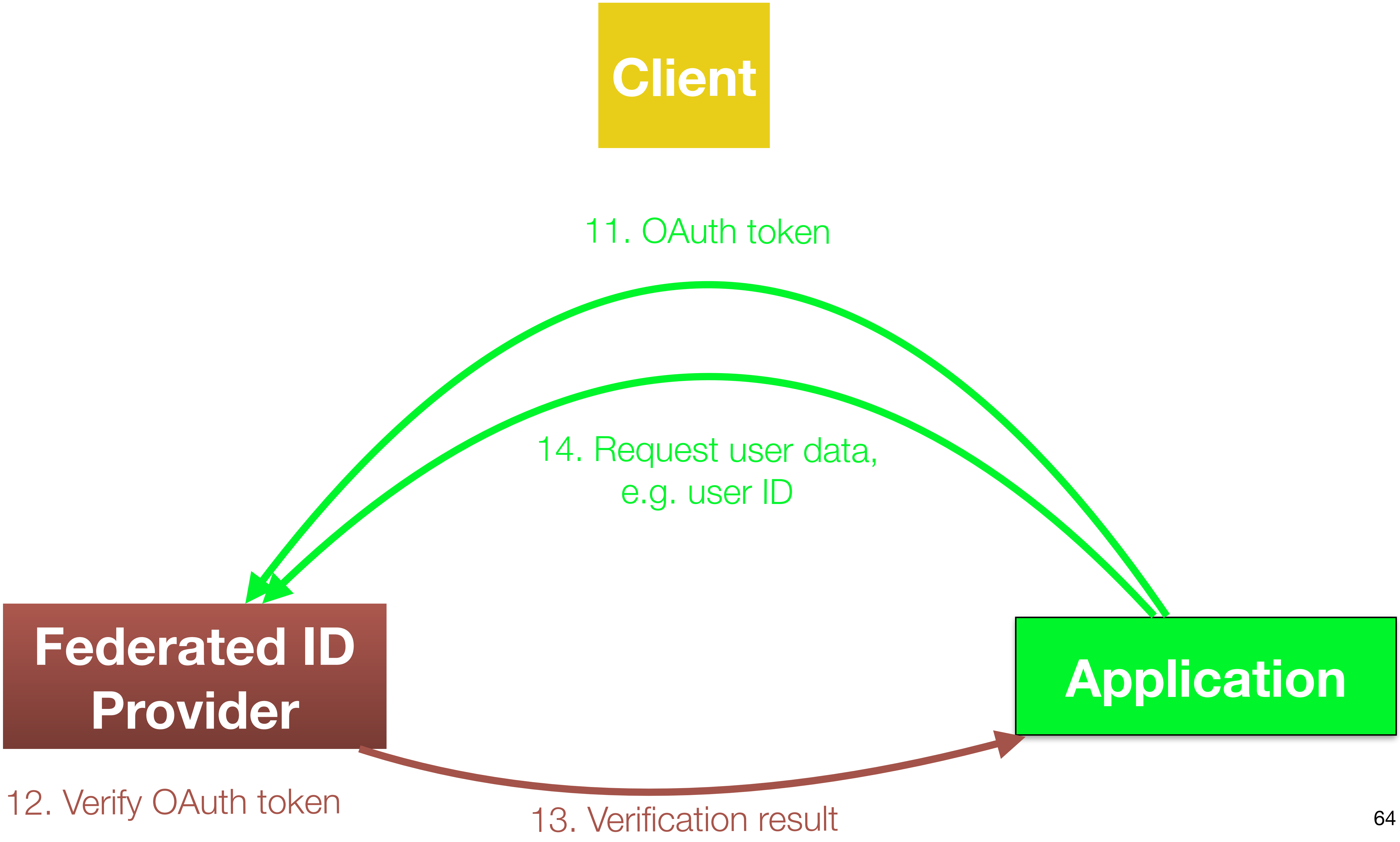


12. Verify OAuth token

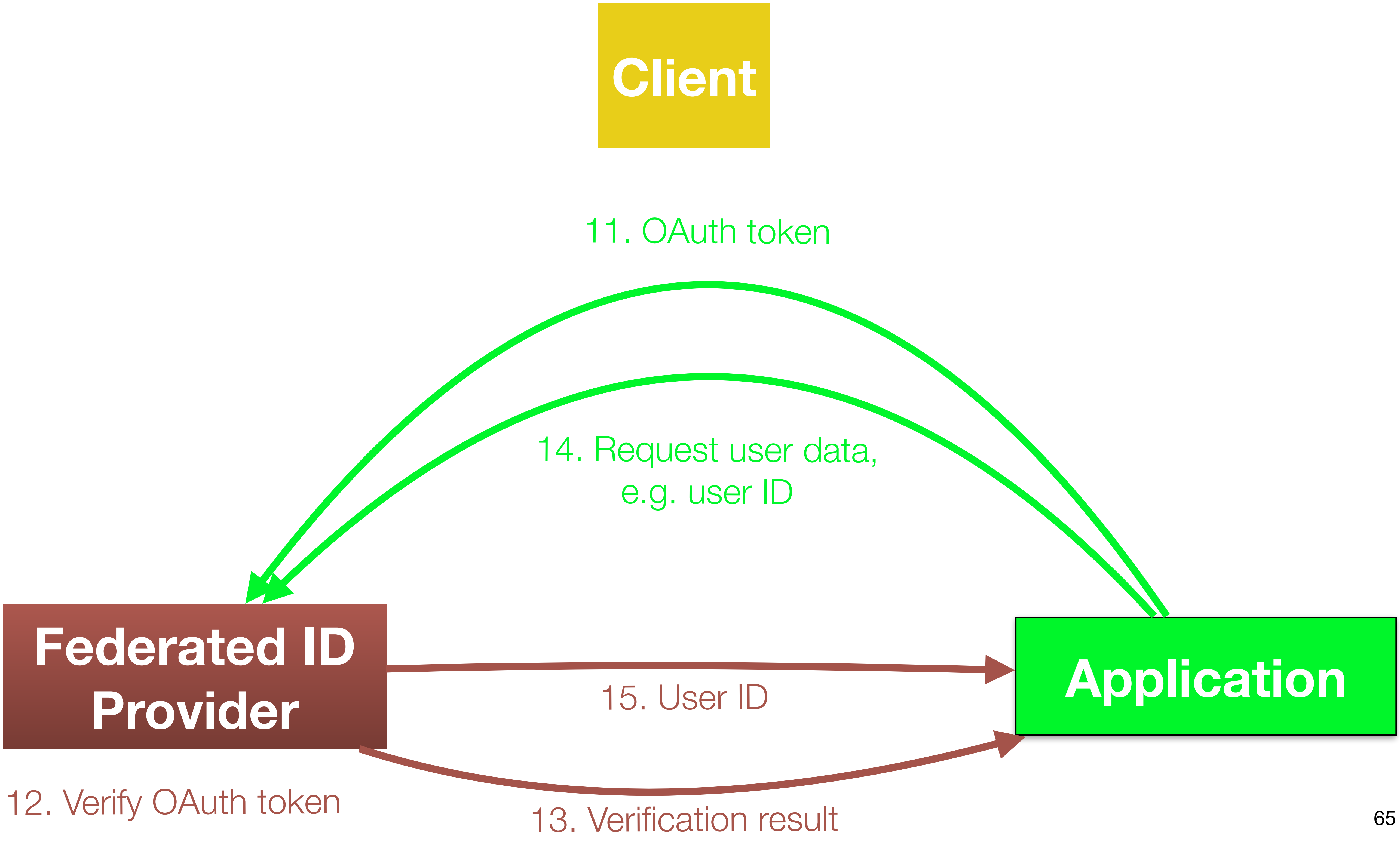
Federated Authentication Interaction



Federated Authentication Interaction



Federated Authentication Interaction



Client

11. OAuth token

14. Request user data, e.g. user ID

Federated ID Provider

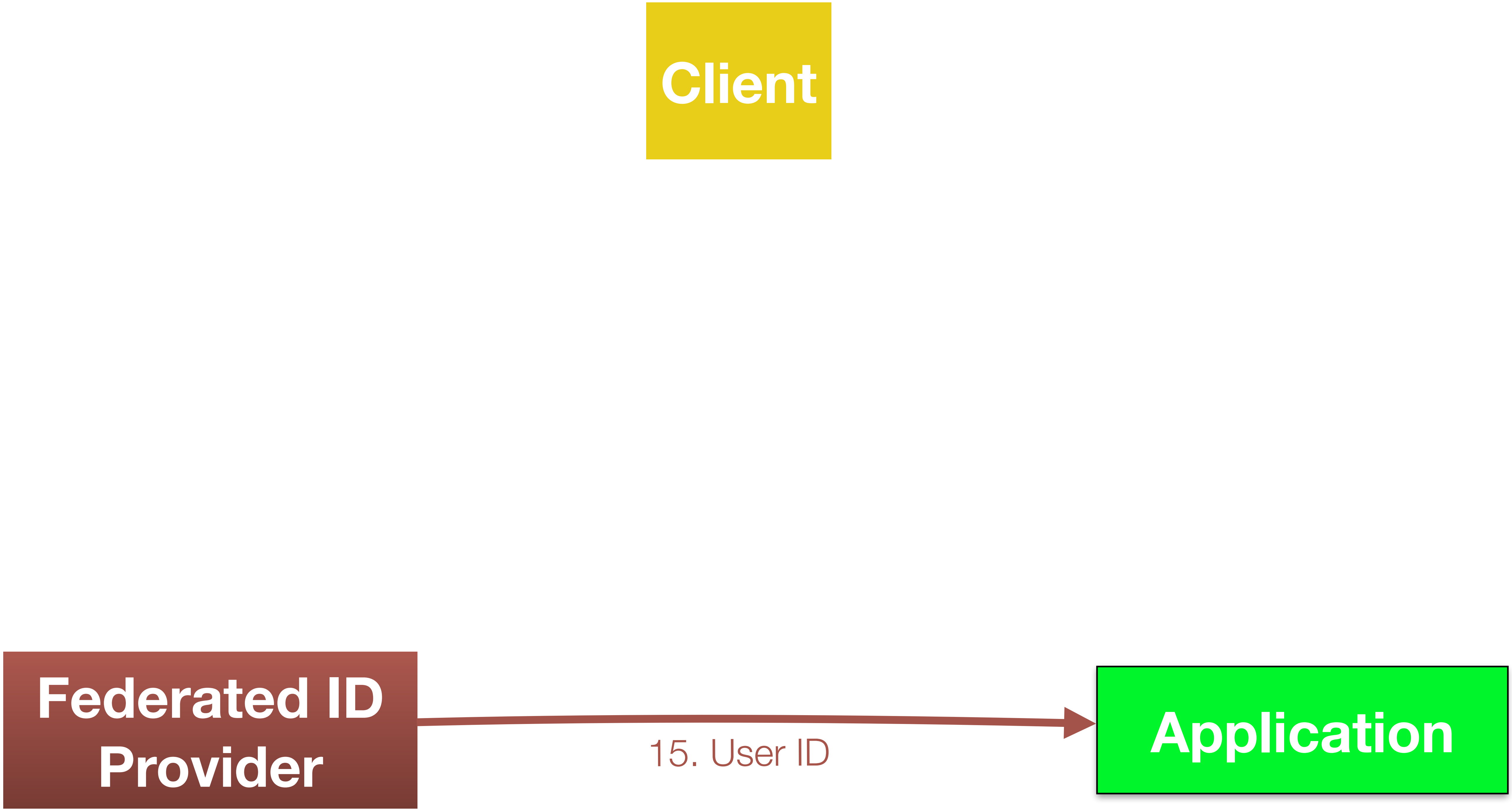
Application

15. User ID

12. Verify OAuth token

13. Verification result

Federated Authentication Interaction



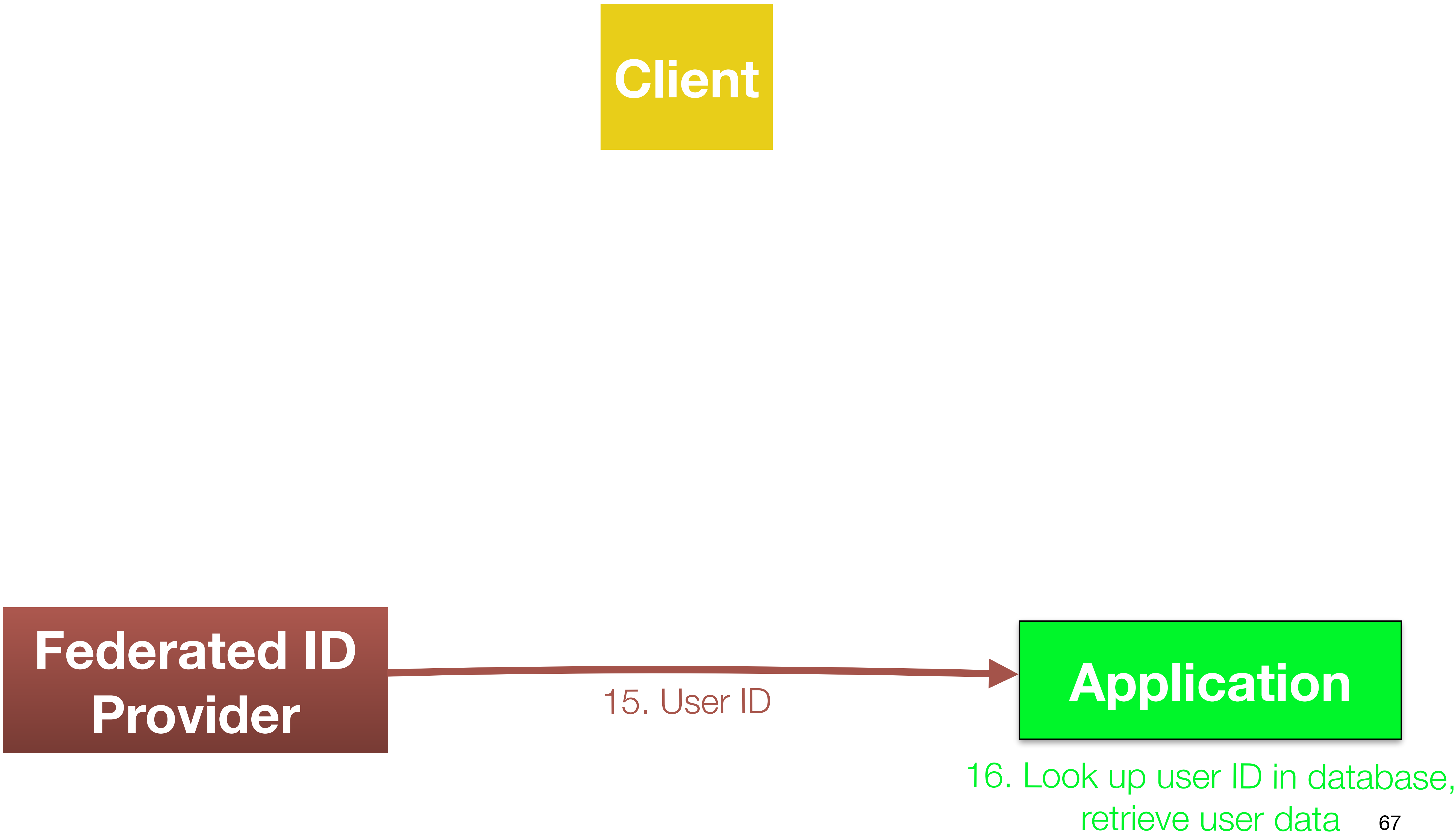
Client

Federated ID Provider

15. User ID

Application

Federated Authentication Interaction



Client

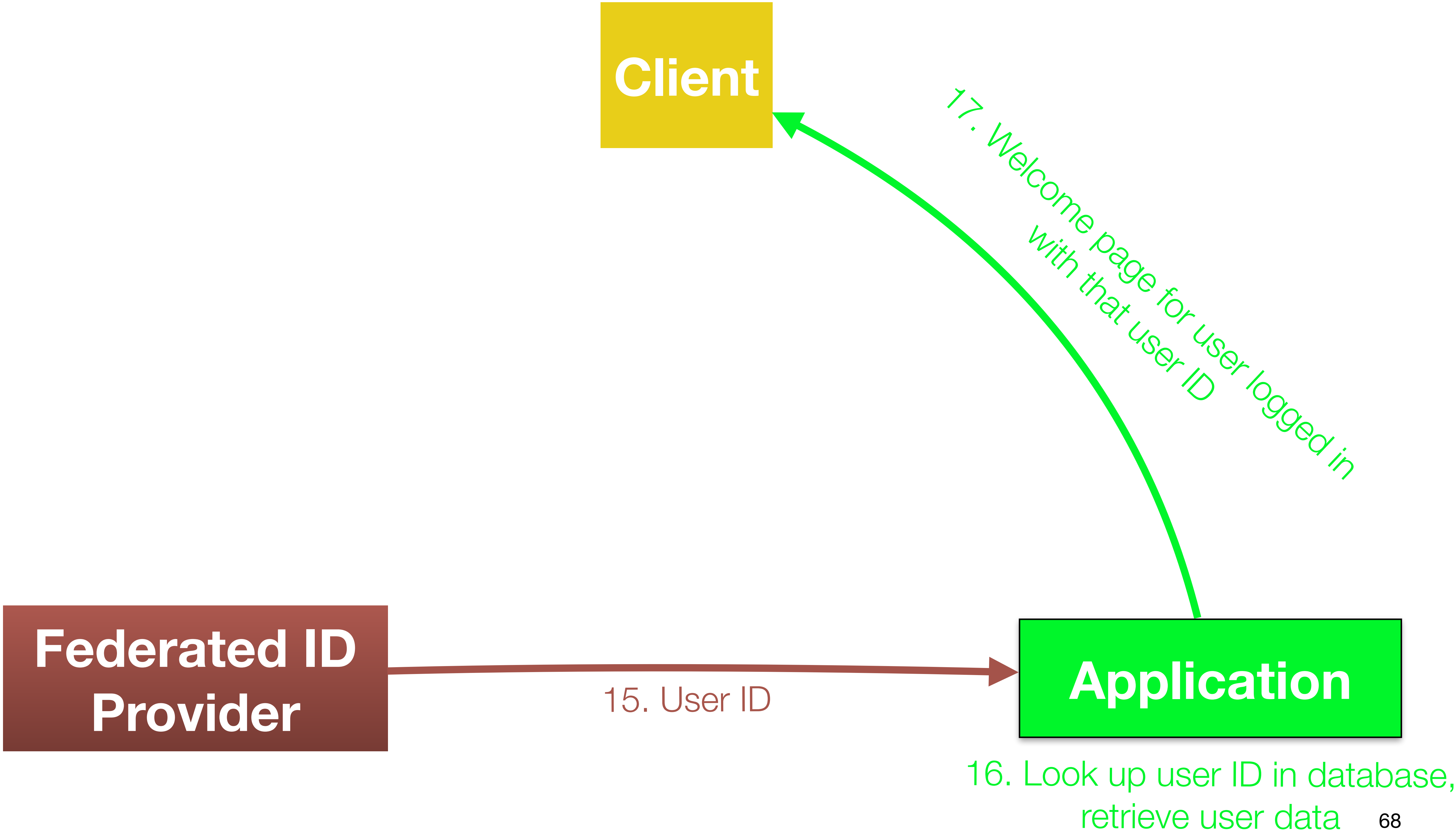
Federated ID Provider

15. User ID

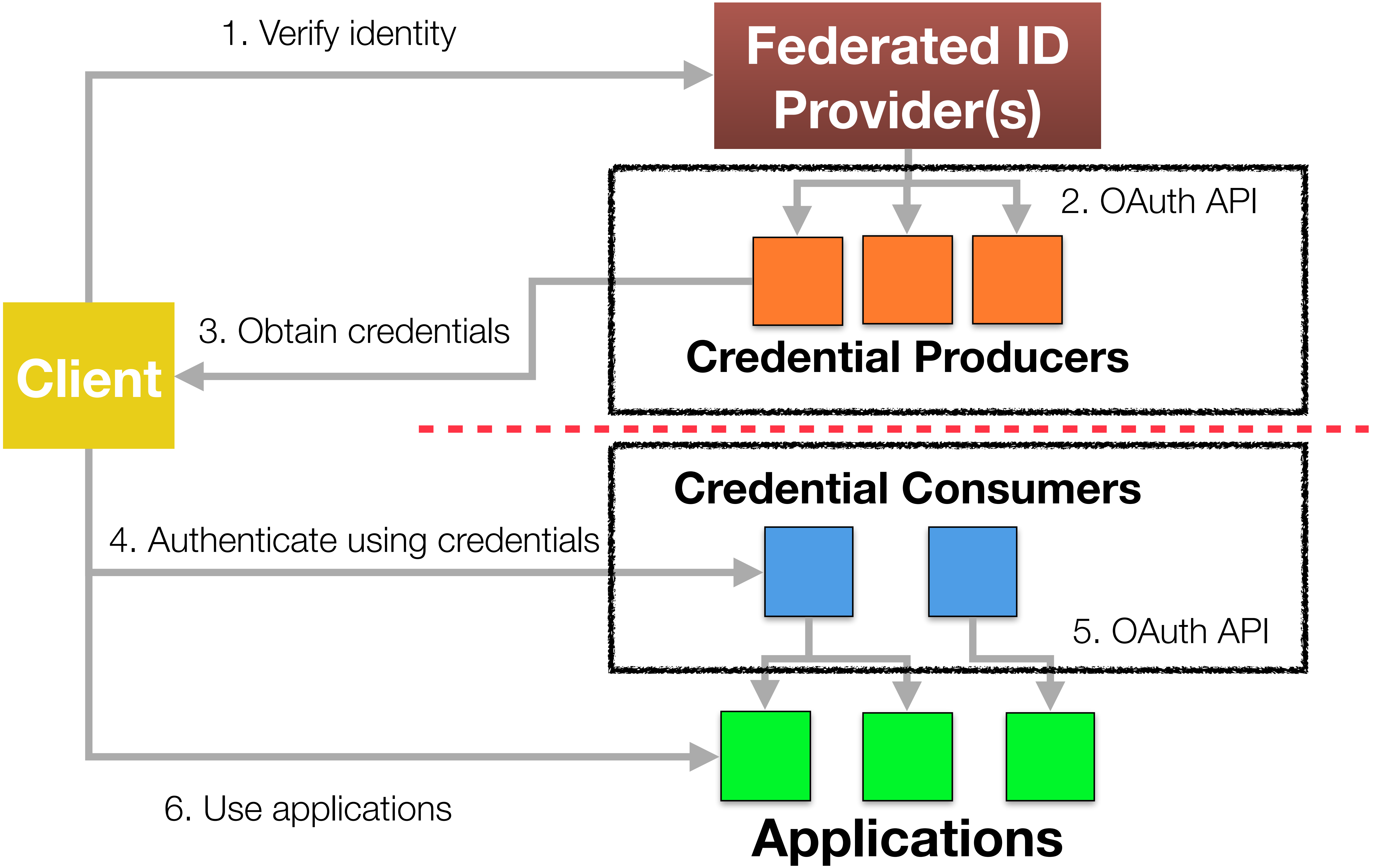
Application

16. Look up user ID in database, retrieve user data

Federated Authentication Interaction


Client
17. Welcome page for user logged in with that user ID
Federated ID Provider
15. User ID
Application
16. Look up user ID in database, retrieve user data

System Architecture



1. Verify identity

Federated ID Provider(s)

2. OAuth API

Credential Producers

3. Obtain credentials

Client

Credential Consumers

4. Authenticate using credentials

5. OAuth API

6. Use applications

Applications

Roadmap

1. Background

2. Work Overview

3. System Architecture

4. **Credential Producers and Consumers**
   - At -Large Credentials
   - Group Credentials

5. Evaluation

6. Conclusions

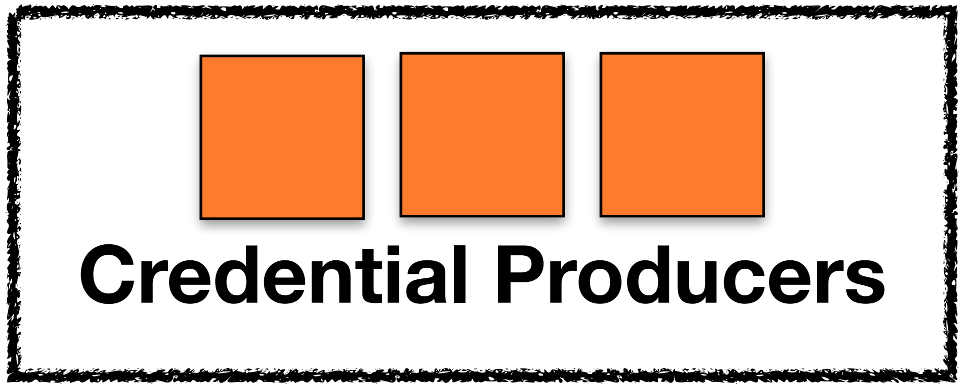
Definition: Privacy Preserving Credential

- **A client uses a privacy preserving credential to prove they own a pseudonym, without revealing their true identity**

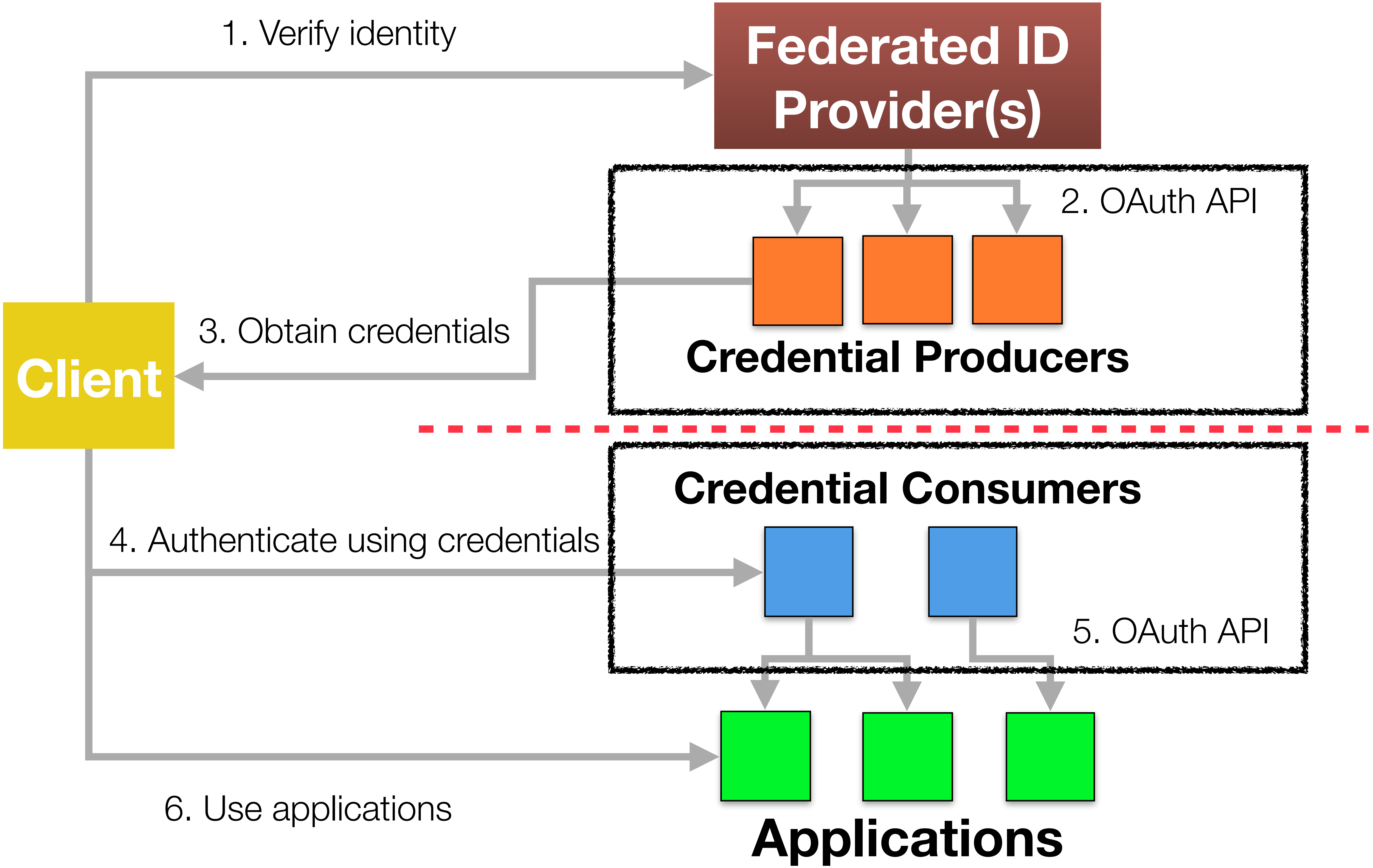
- Using privacy preserving cryptographic techniques

Credential Producers

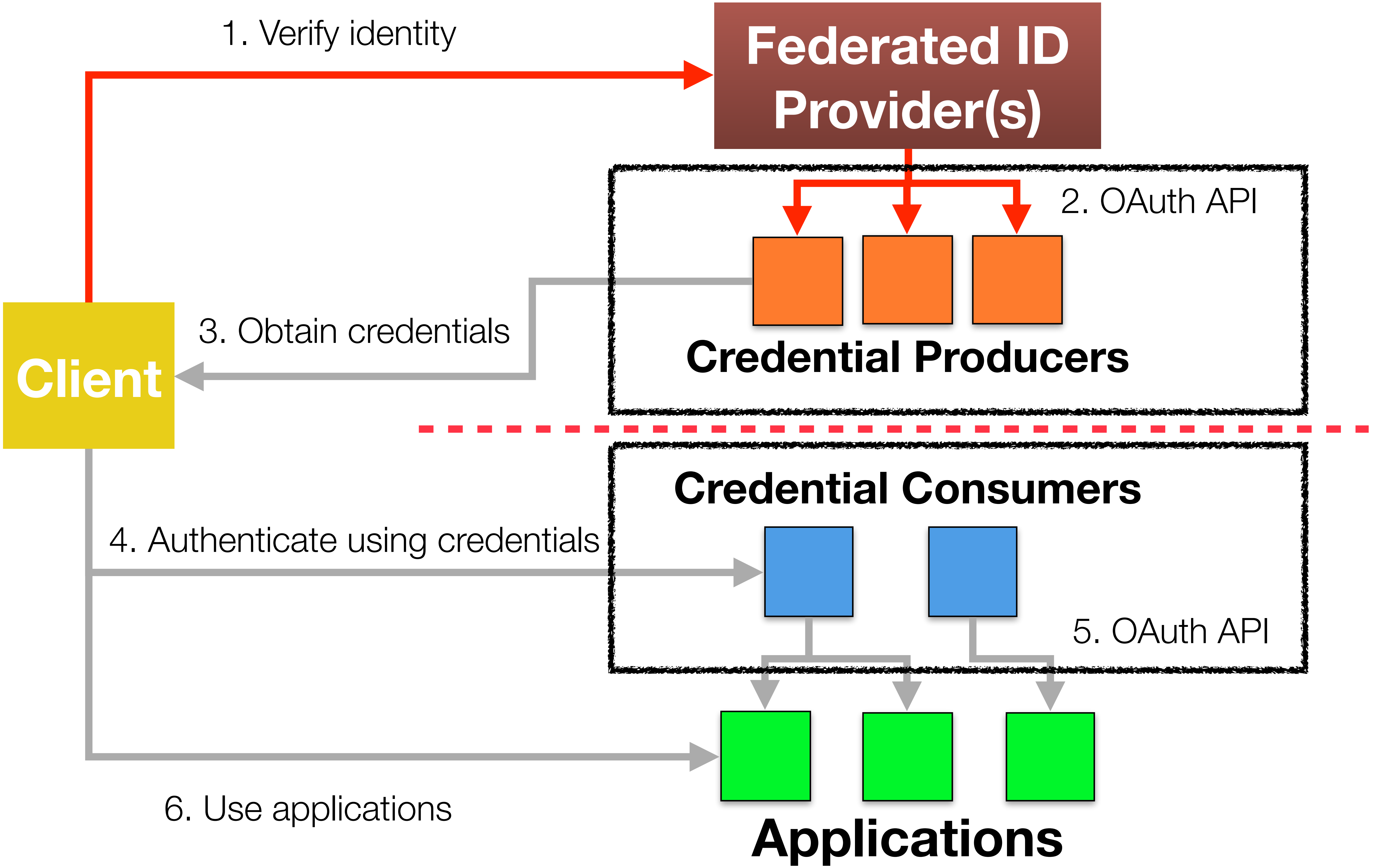
- Several credential producer servers collectively act to assign credentials to clients
- (t,n) threshold model - t of n servers can collectively assign a credential to a client
- Acts as an “application” in OAuth protocol to authenticate client with federated ID provider



System Architecture

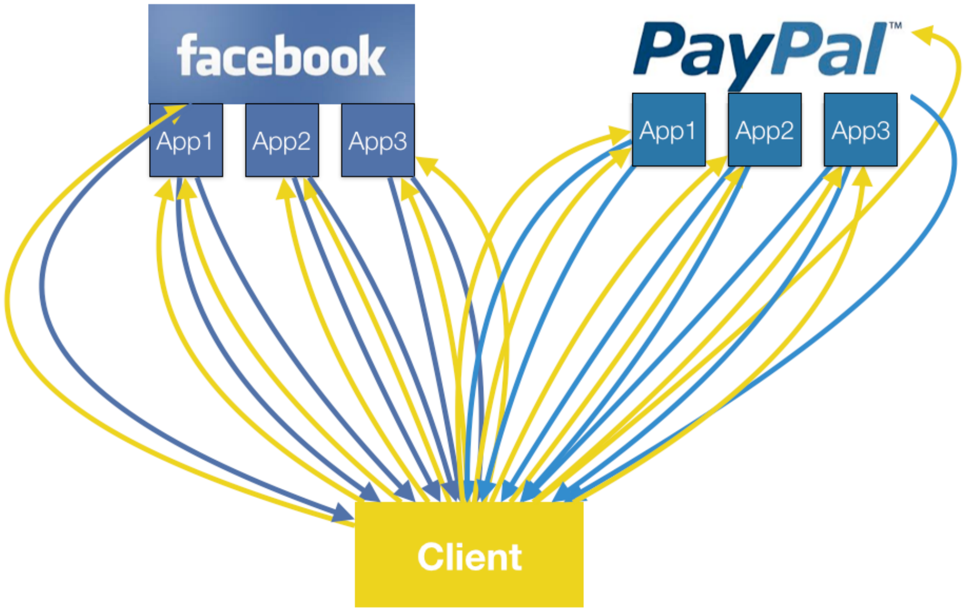


System Architecture

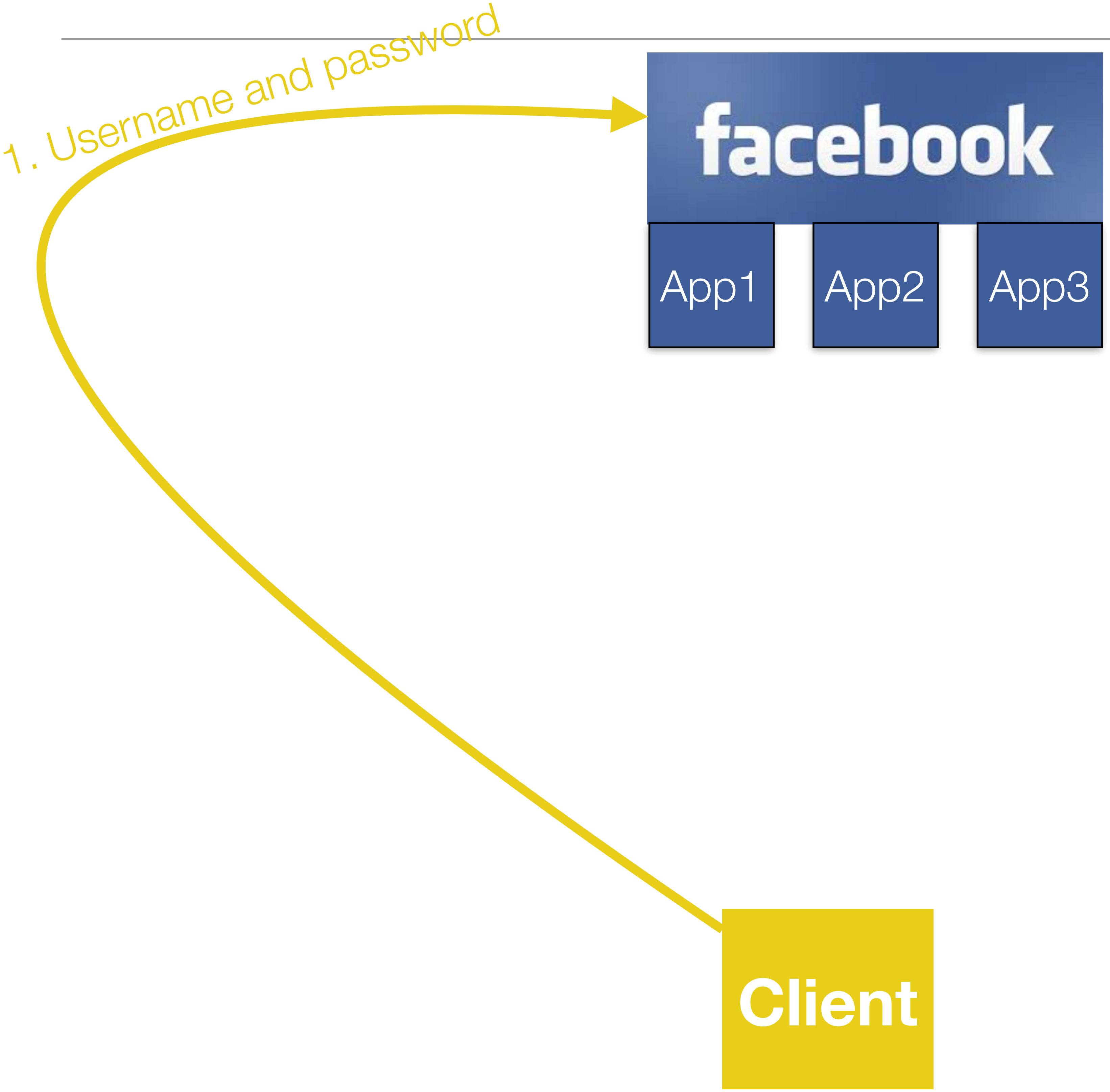


Credential Assignment Mechanism

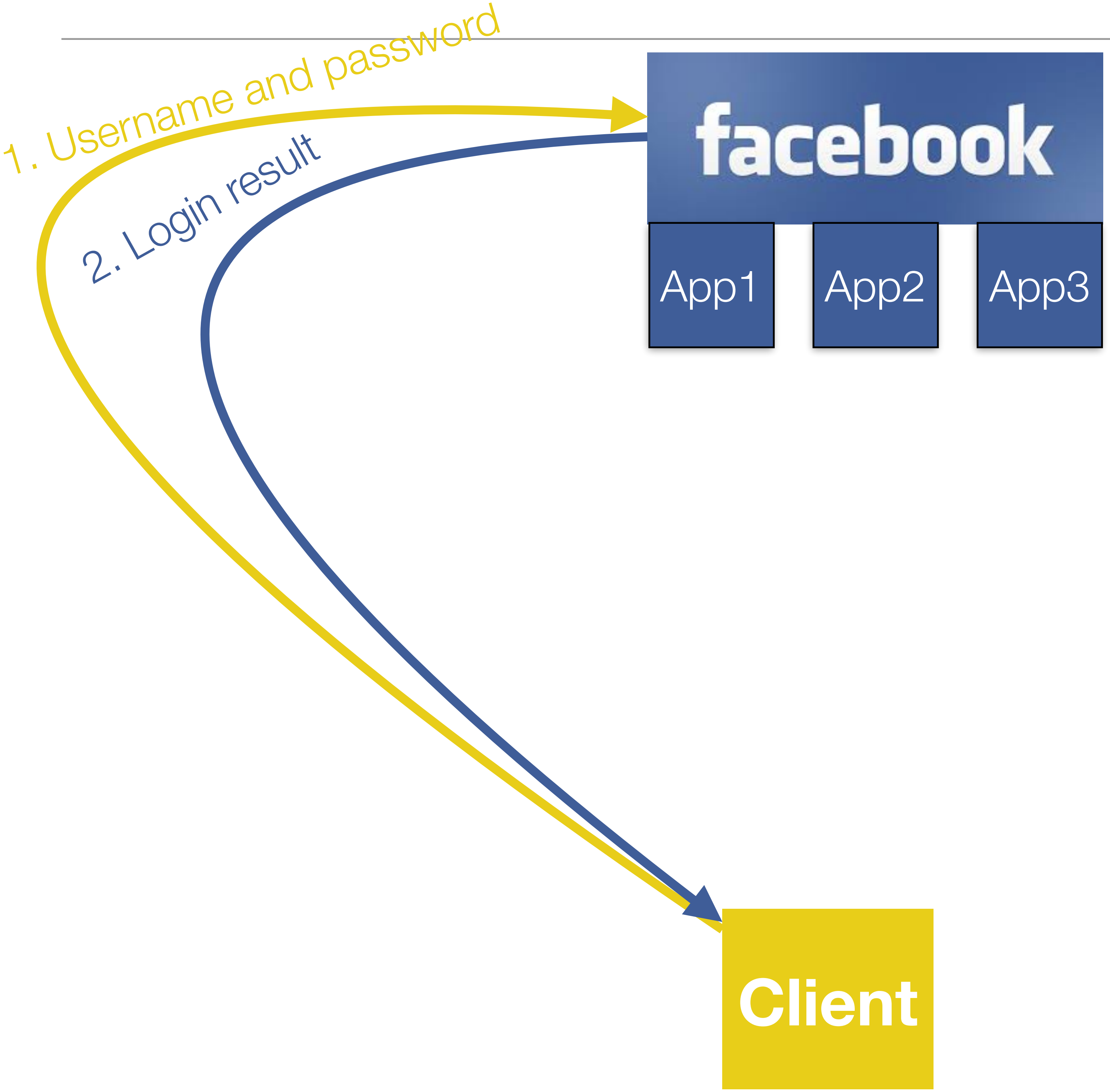
Obtaining OAuth tokens



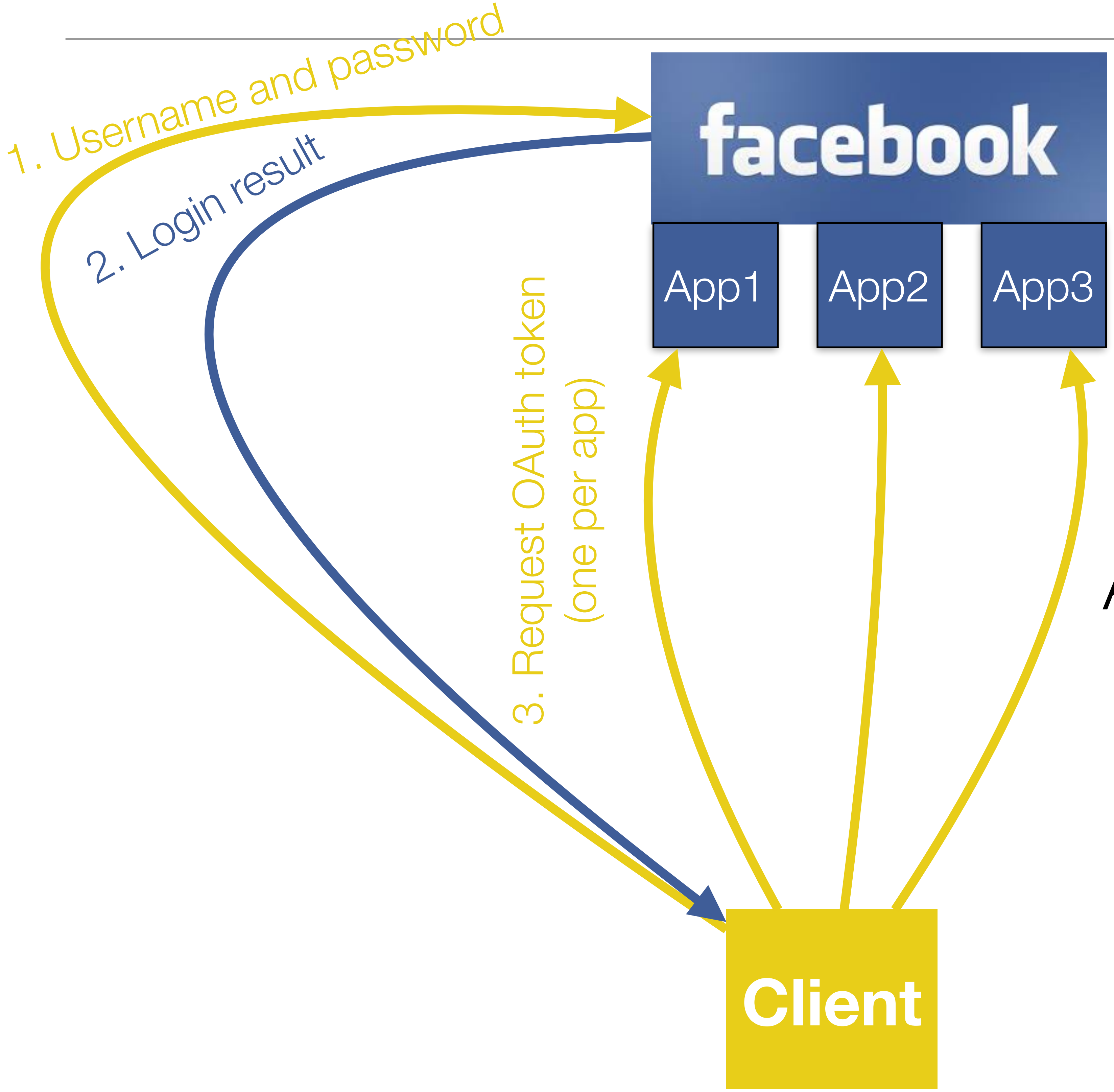
Credential Assignment Mechanism



Credential Assignment Mechanism

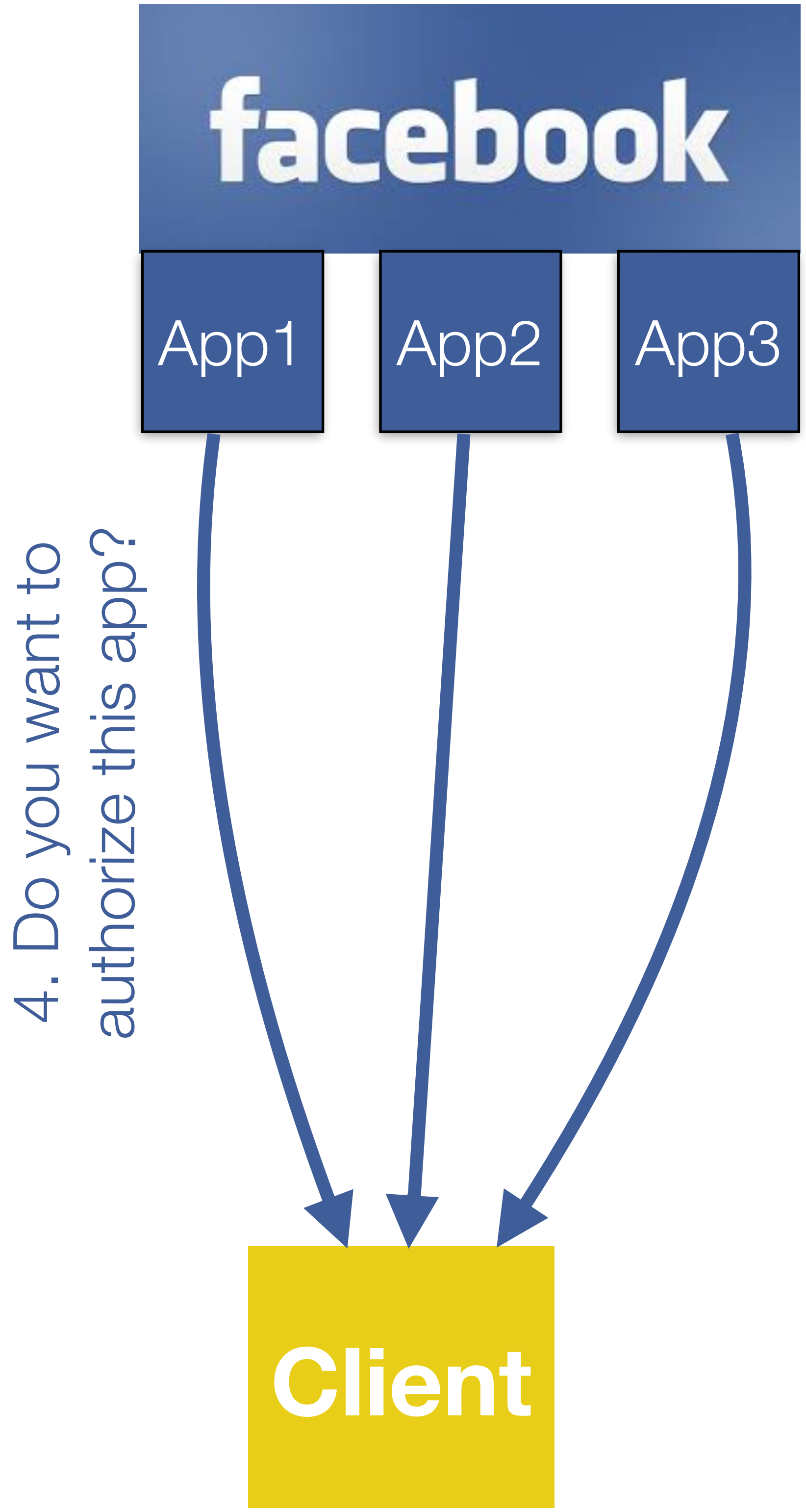


Credential Assignment Mechanism



Requests performed in parallel.
Automated by a Chrome extension so user does not have to manually repeat the same task.

Credential Assignment Mechanism



facebook

App1

App2

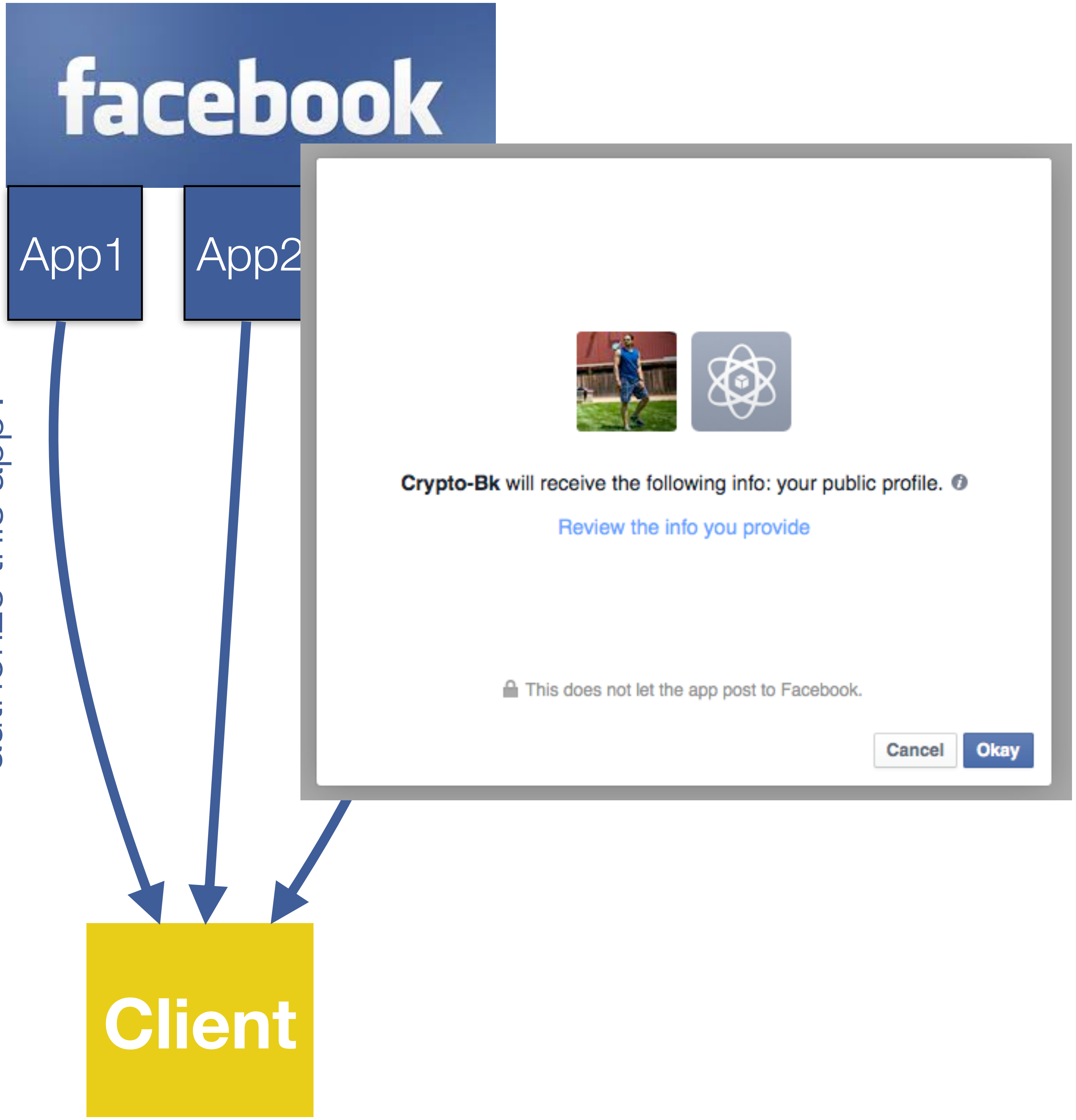
App3

4. Do you want to authorize this app?

Client

Credential Assignment Mechanism

4. Do you want to authorize this app?

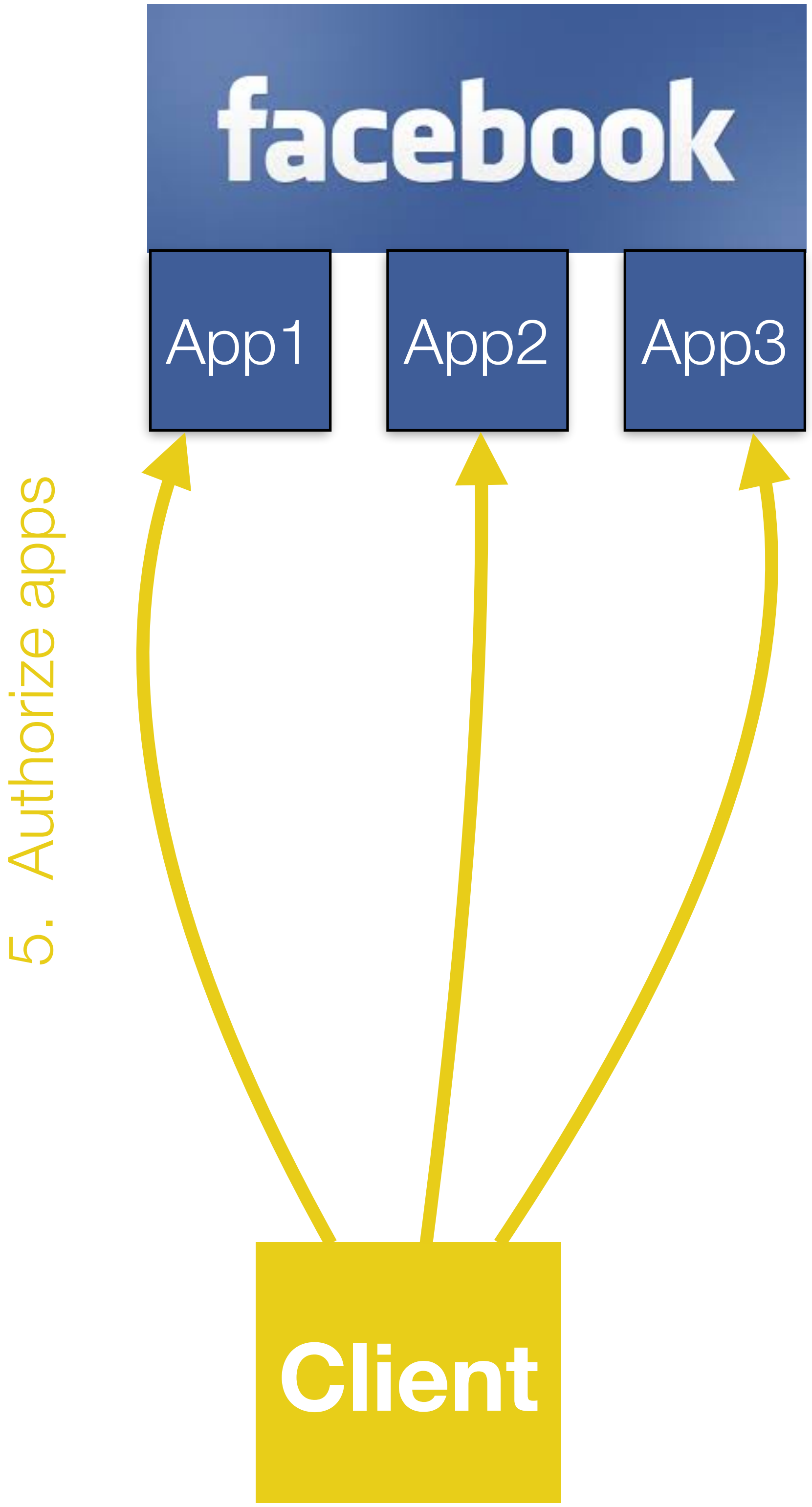


Crypto-Bk will receive the following info: your public profile.

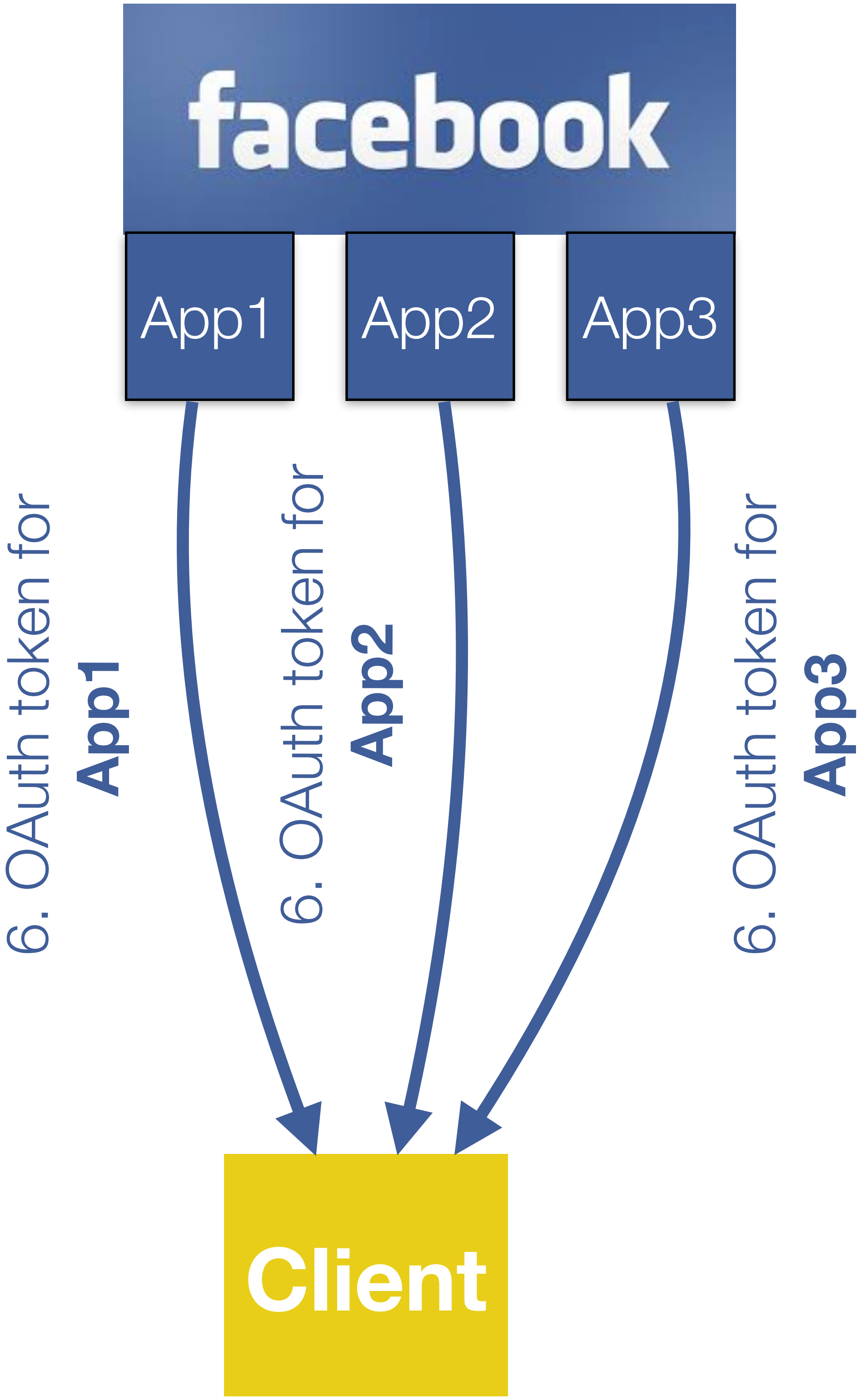
Review the info you provide

This does not let the app post to Facebook.

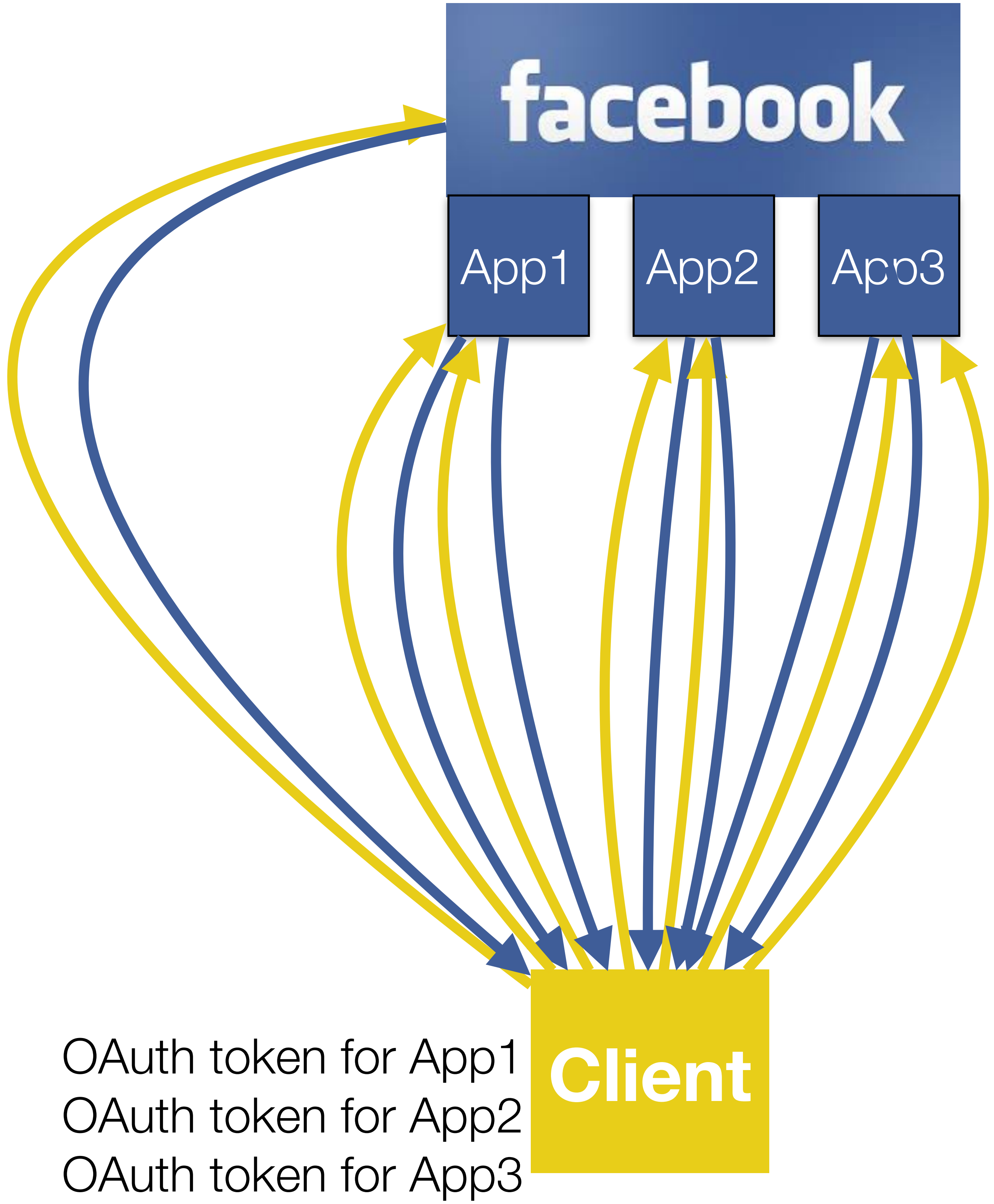
Credential Assignment Mechanism



Credential Assignment Mechanism



Credential Assignment Mechanism



facebook

App1

App2

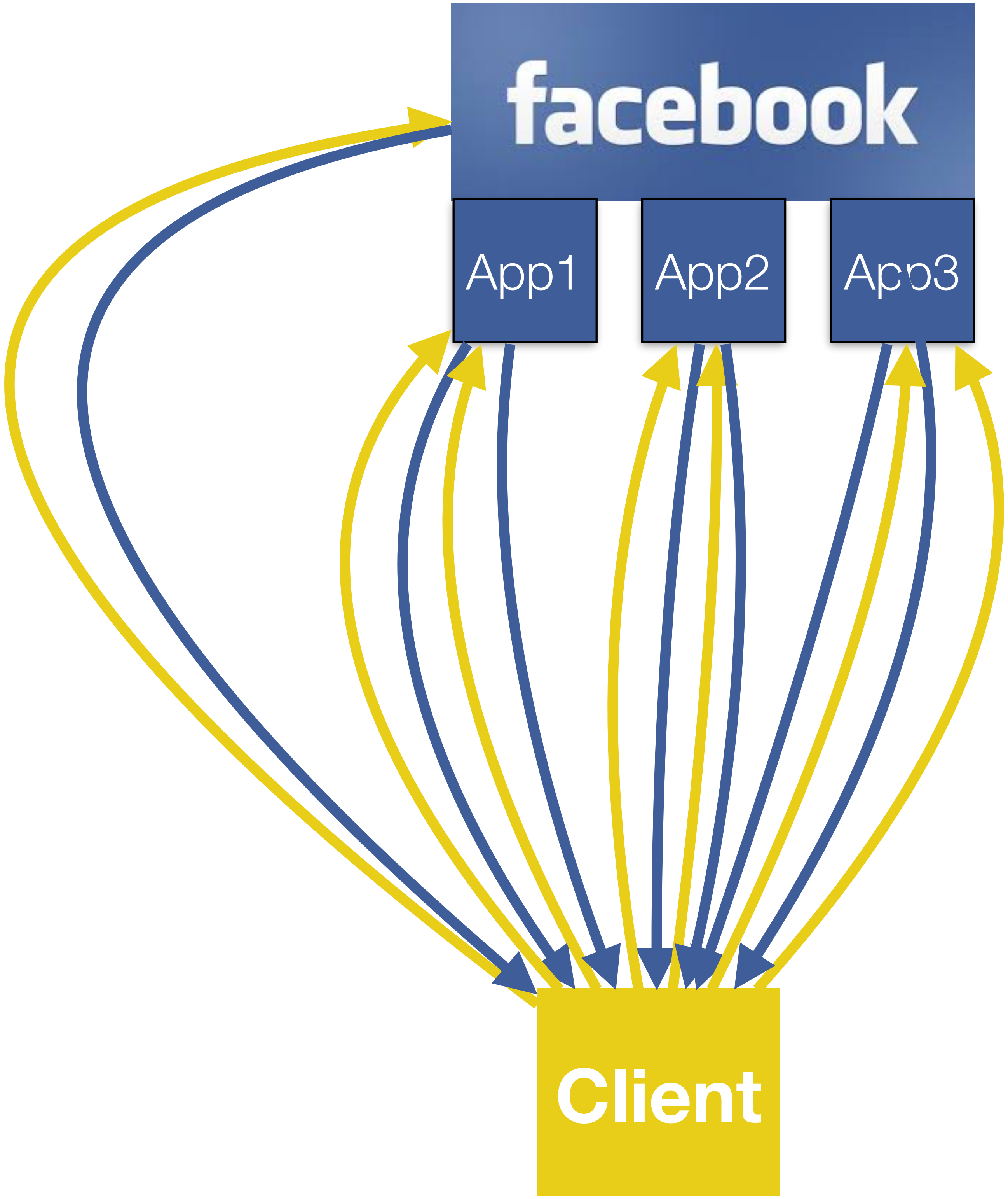
App3

Client

OAuth token for App1
OAuth token for App2
OAuth token for App3

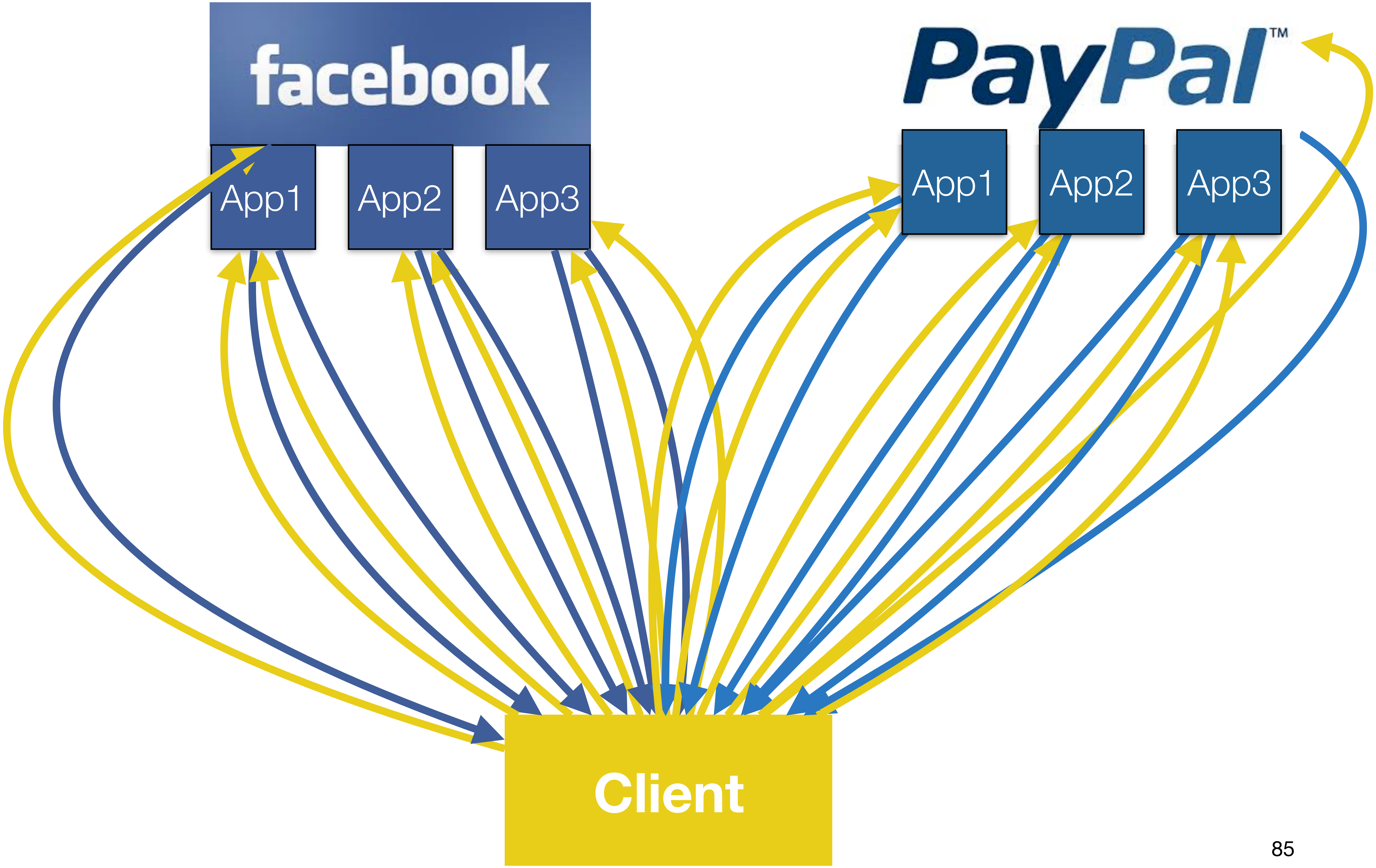
Client now has one
OAuth token per app.
Each app corresponds
to one credential
producer server.

Credential Assignment Mechanism

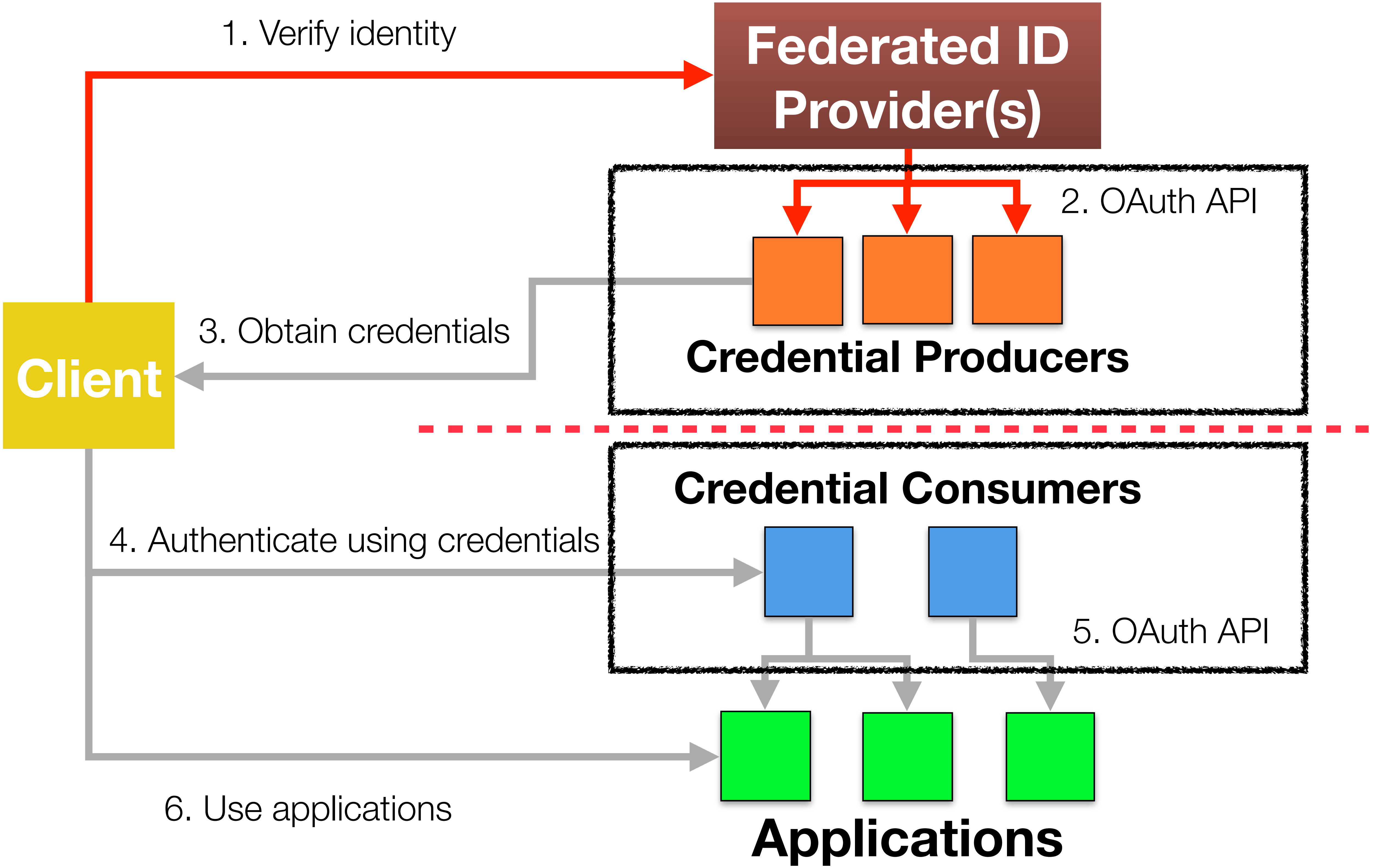


Multiple ID provider use case: This process is performed for each federated ID provider. The user only has to enter their username and password once per federated ID provider. The other steps are automated by a Chrome extension.

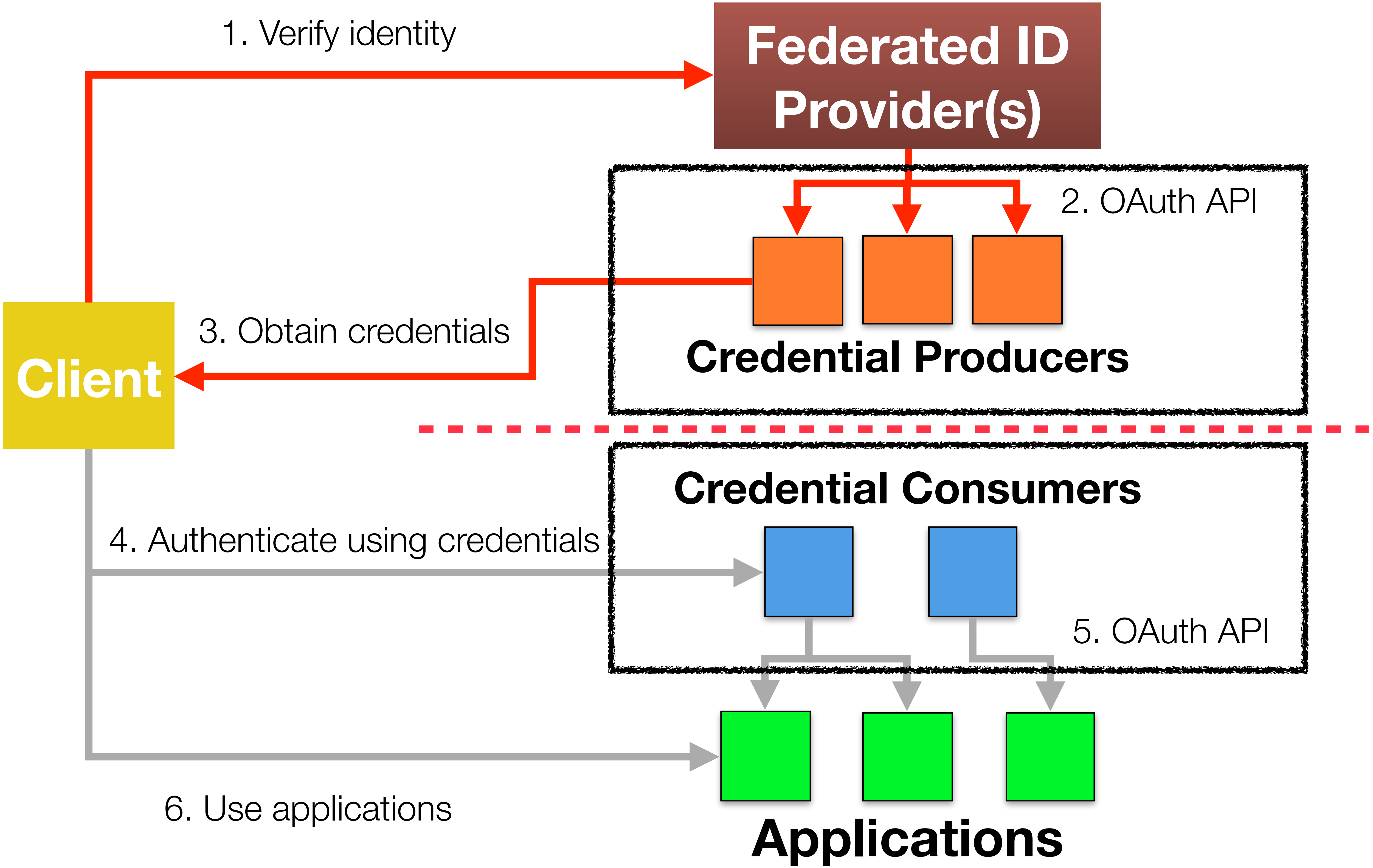
Credential Assignment Mechanism



System Architecture

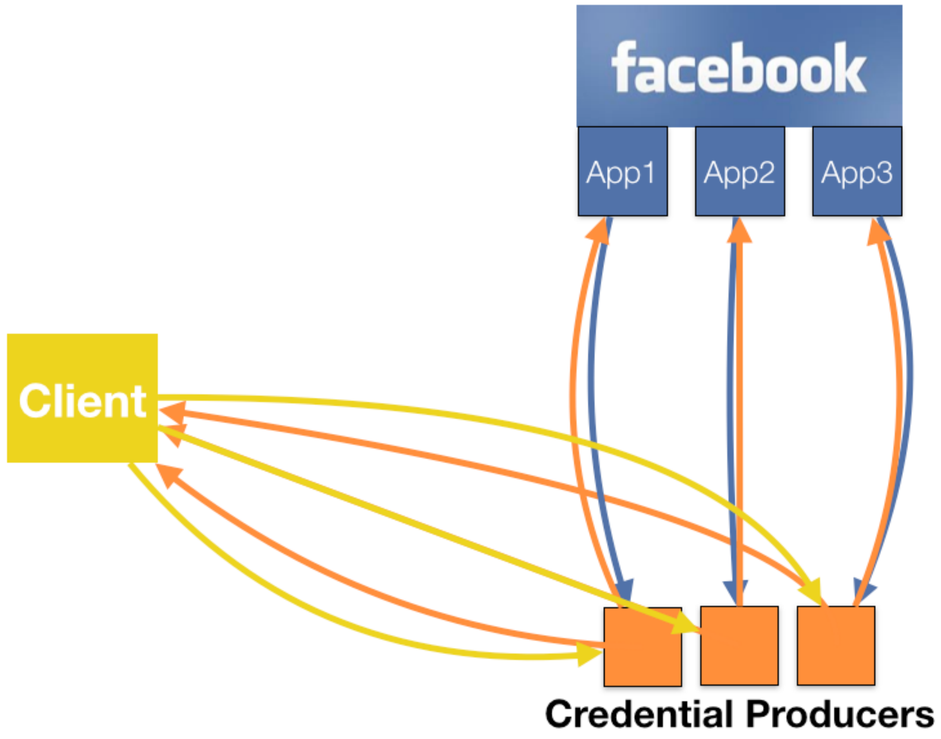


System Architecture

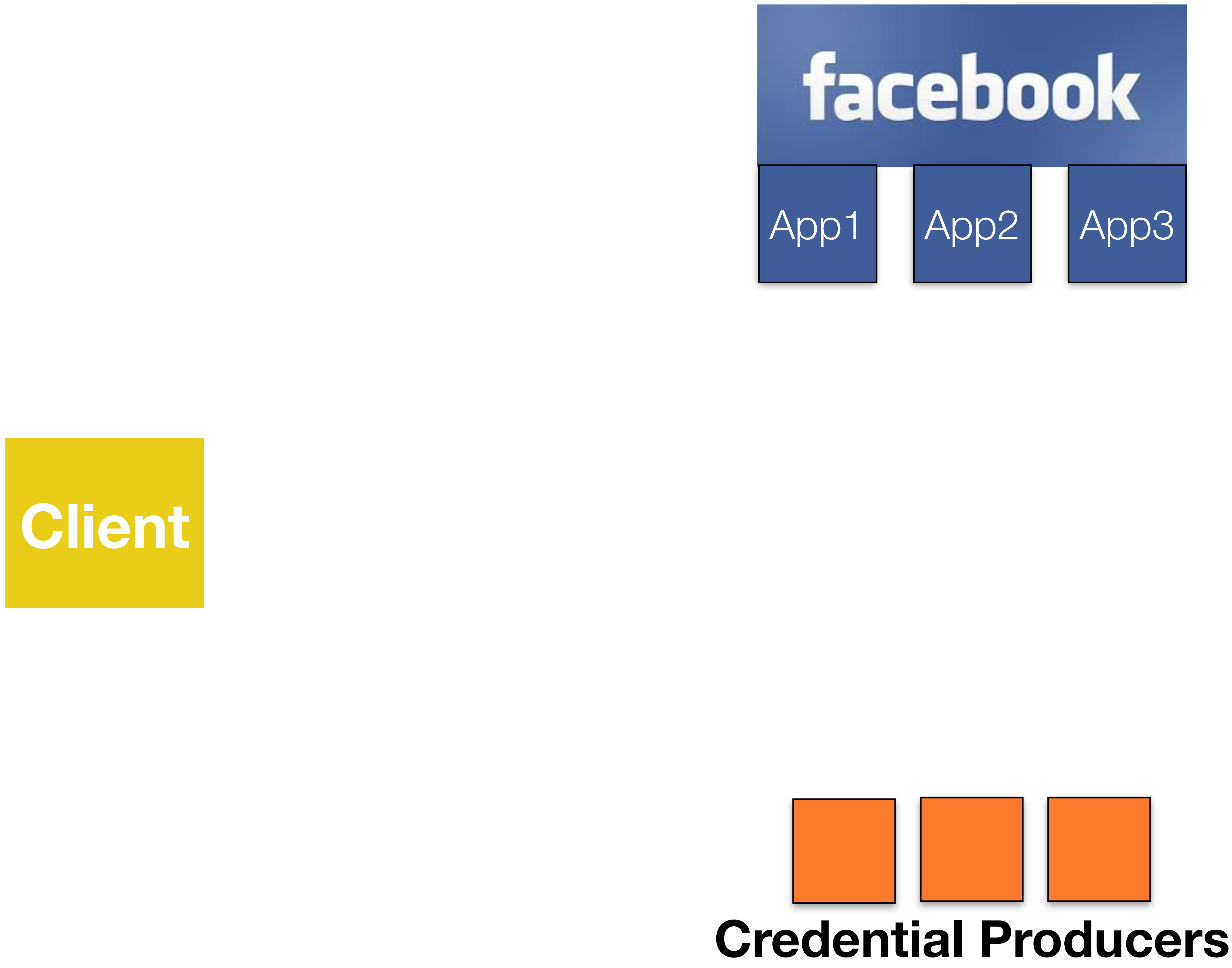


Credential Assignment Mechanism

Obtaining credentials



Credential Assignment Mechanism



facebook

App1

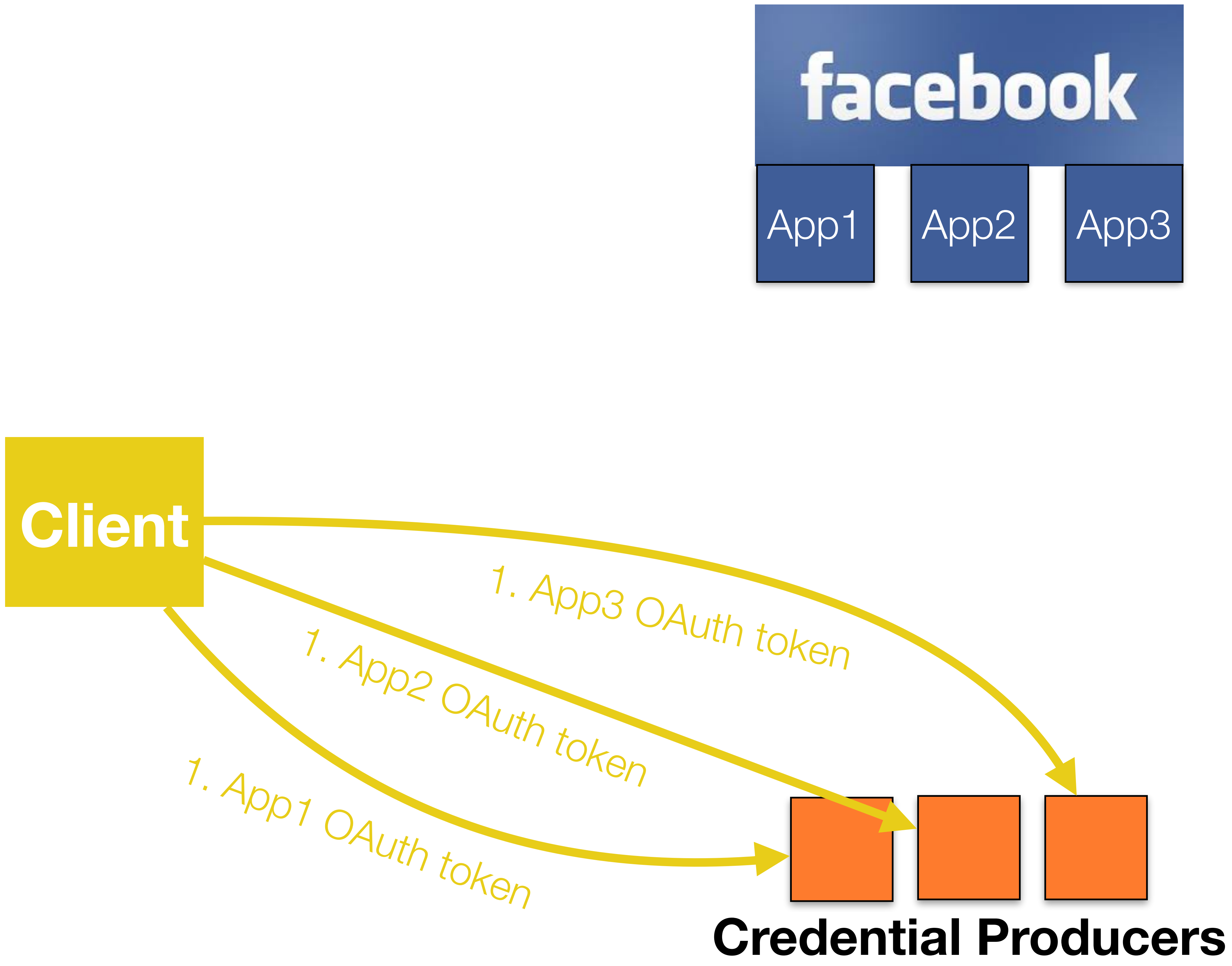
App2

App3

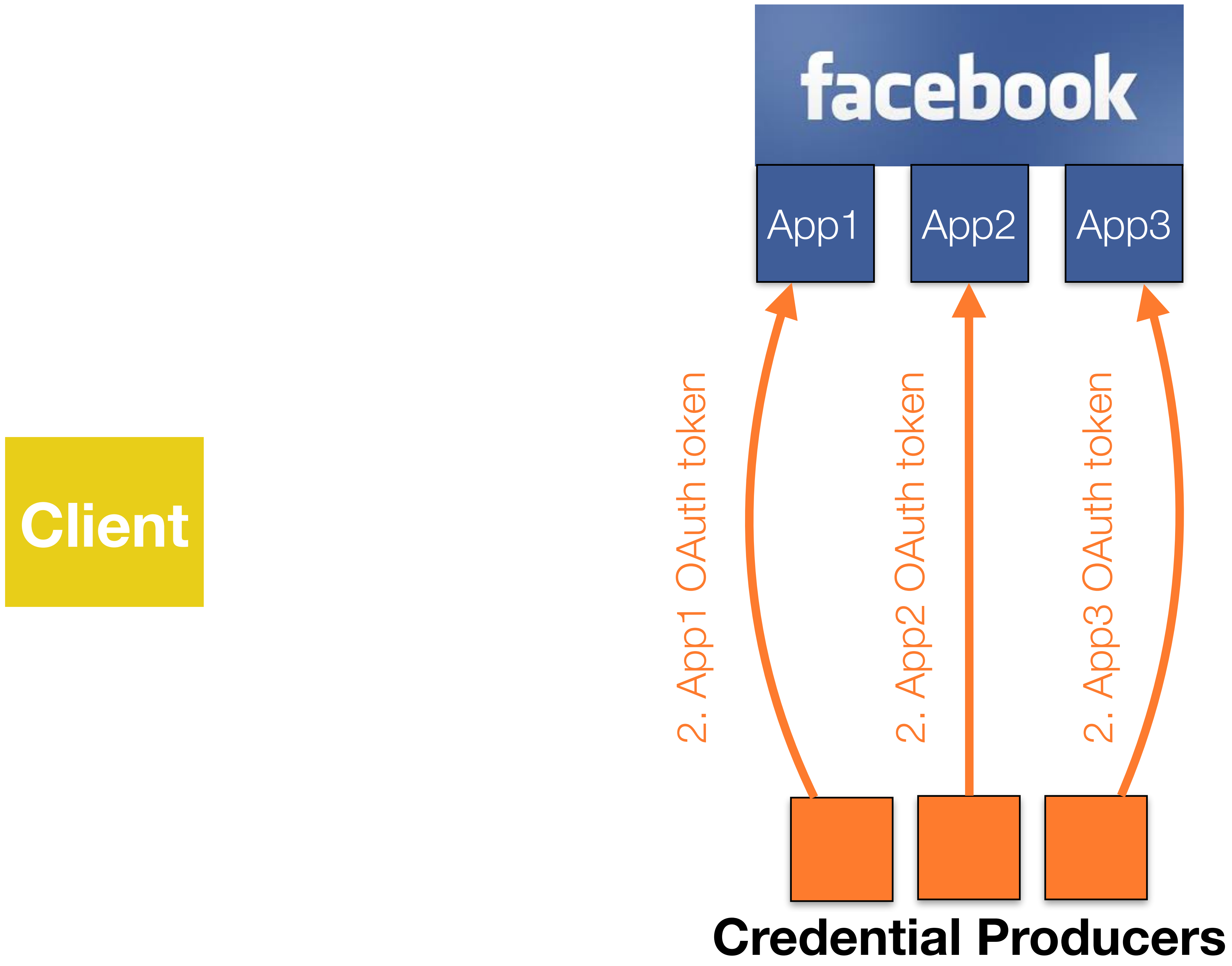
Client

Credential Producers

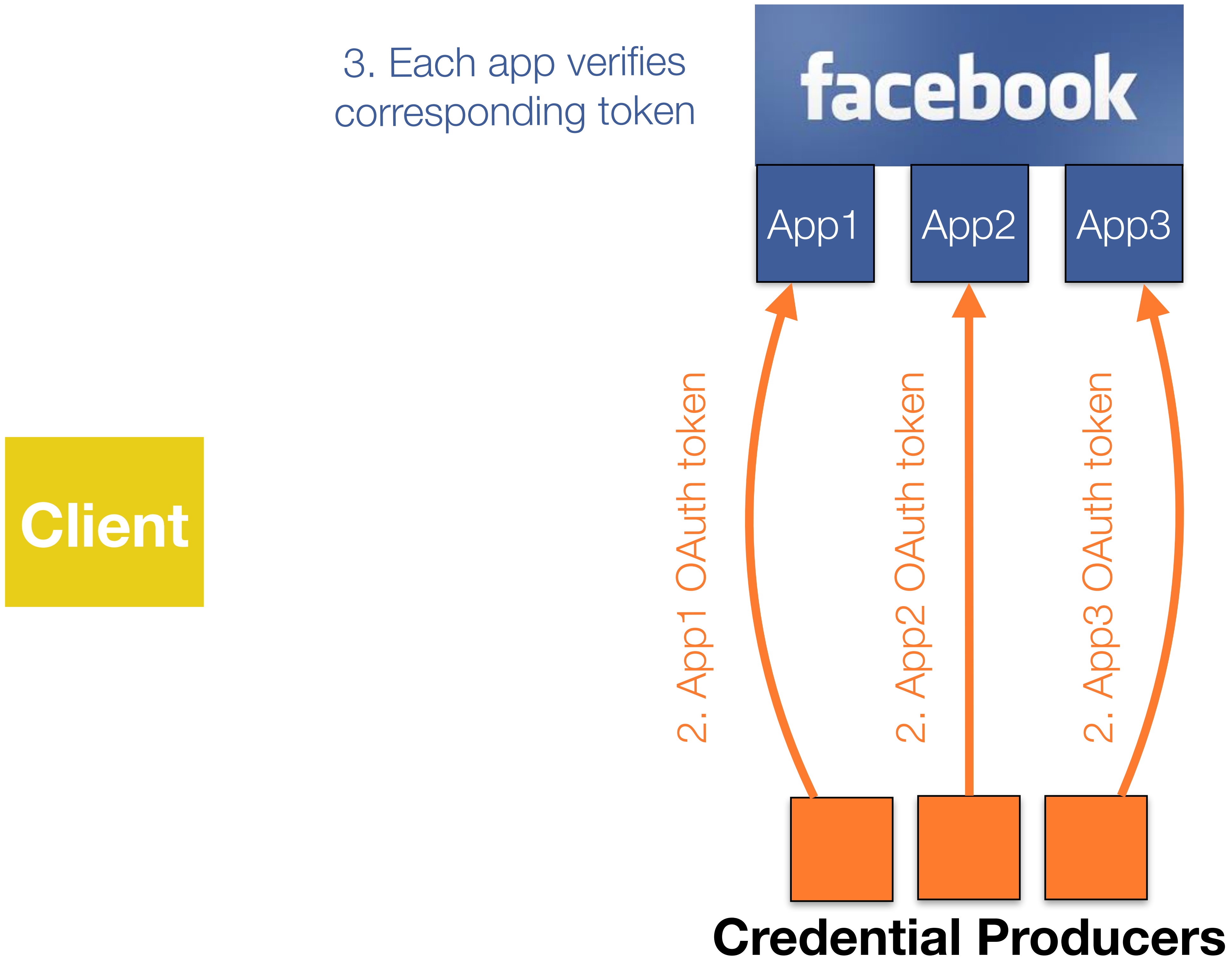
Credential Assignment Mechanism



Credential Assignment Mechanism



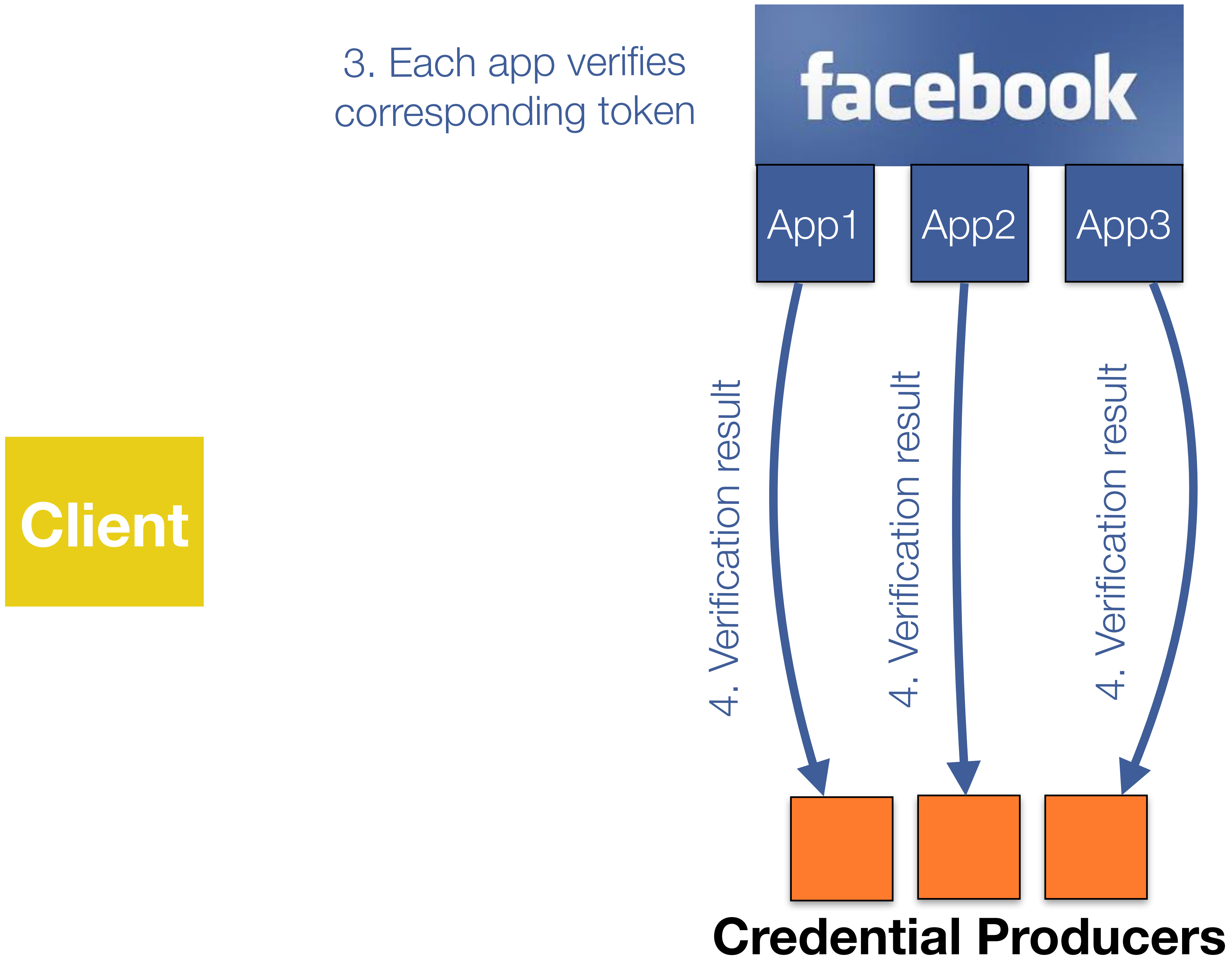
Credential Assignment Mechanism



3. Each app verifies corresponding token

Credential Producers

Credential Assignment Mechanism



3. Each app verifies corresponding token

facebook

App1

App2

App3

4. Verification result

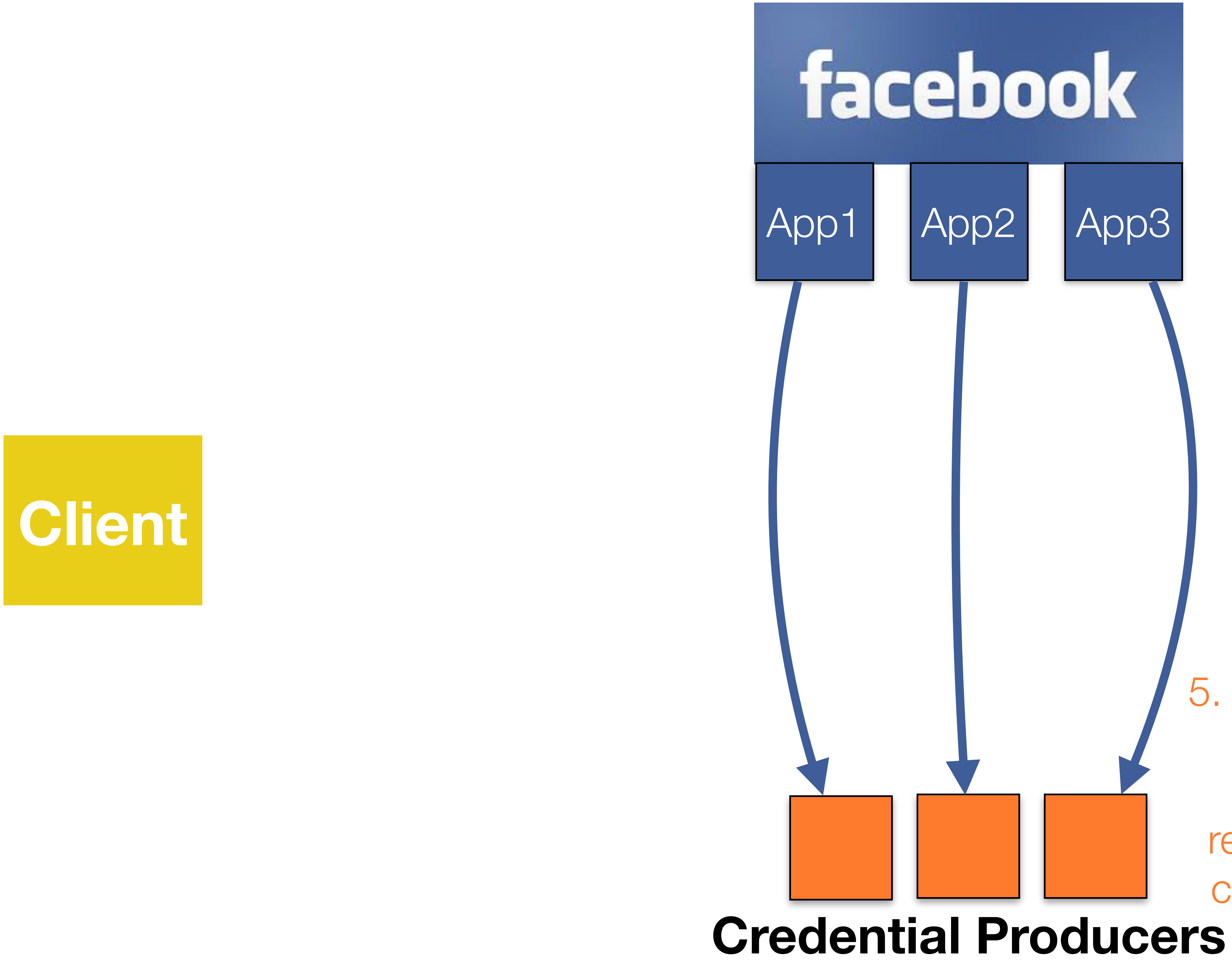
4. Verification result

4. Verification result

Client

Credential Producers

Credential Assignment Mechanism



App1

App2

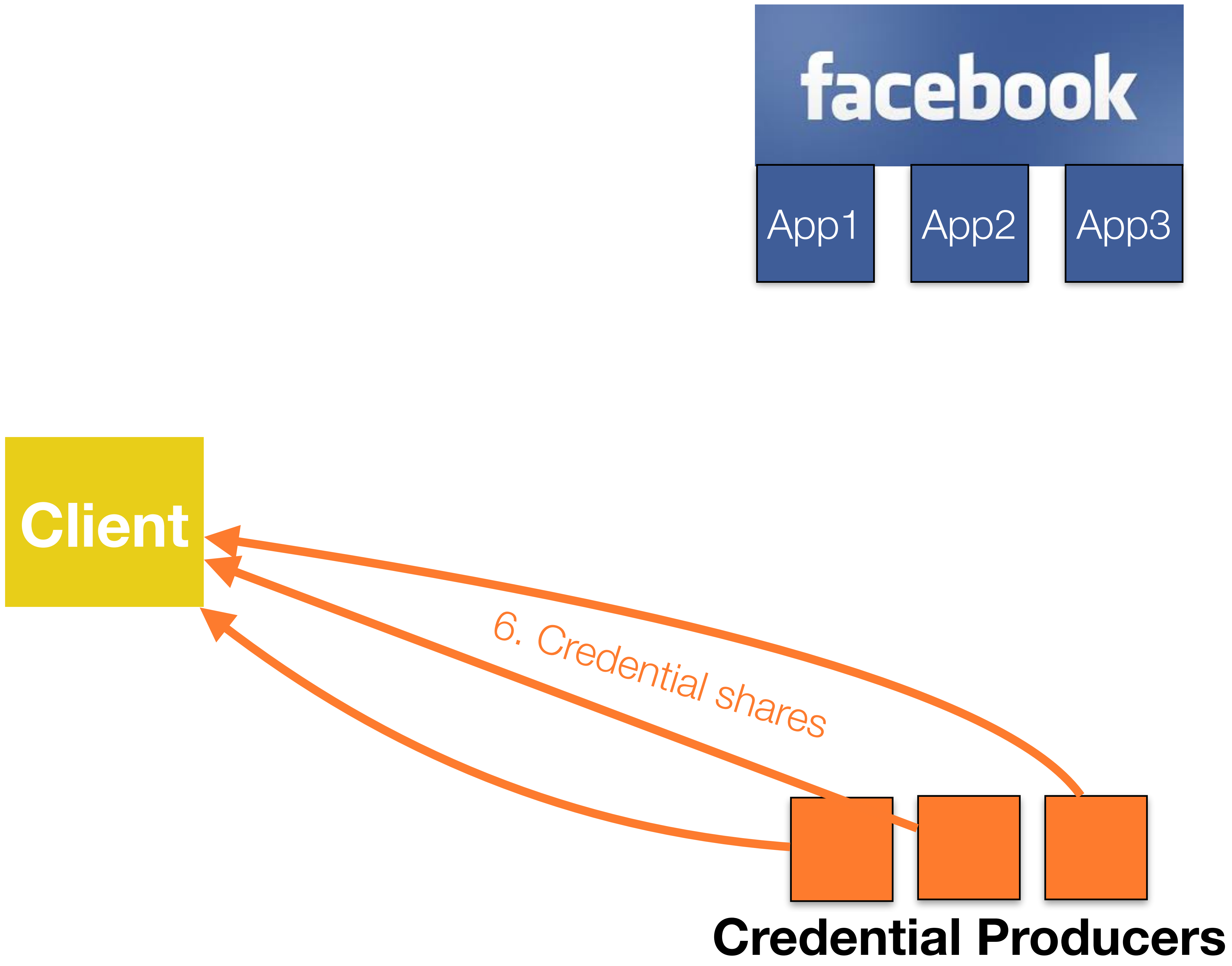
App3

Client

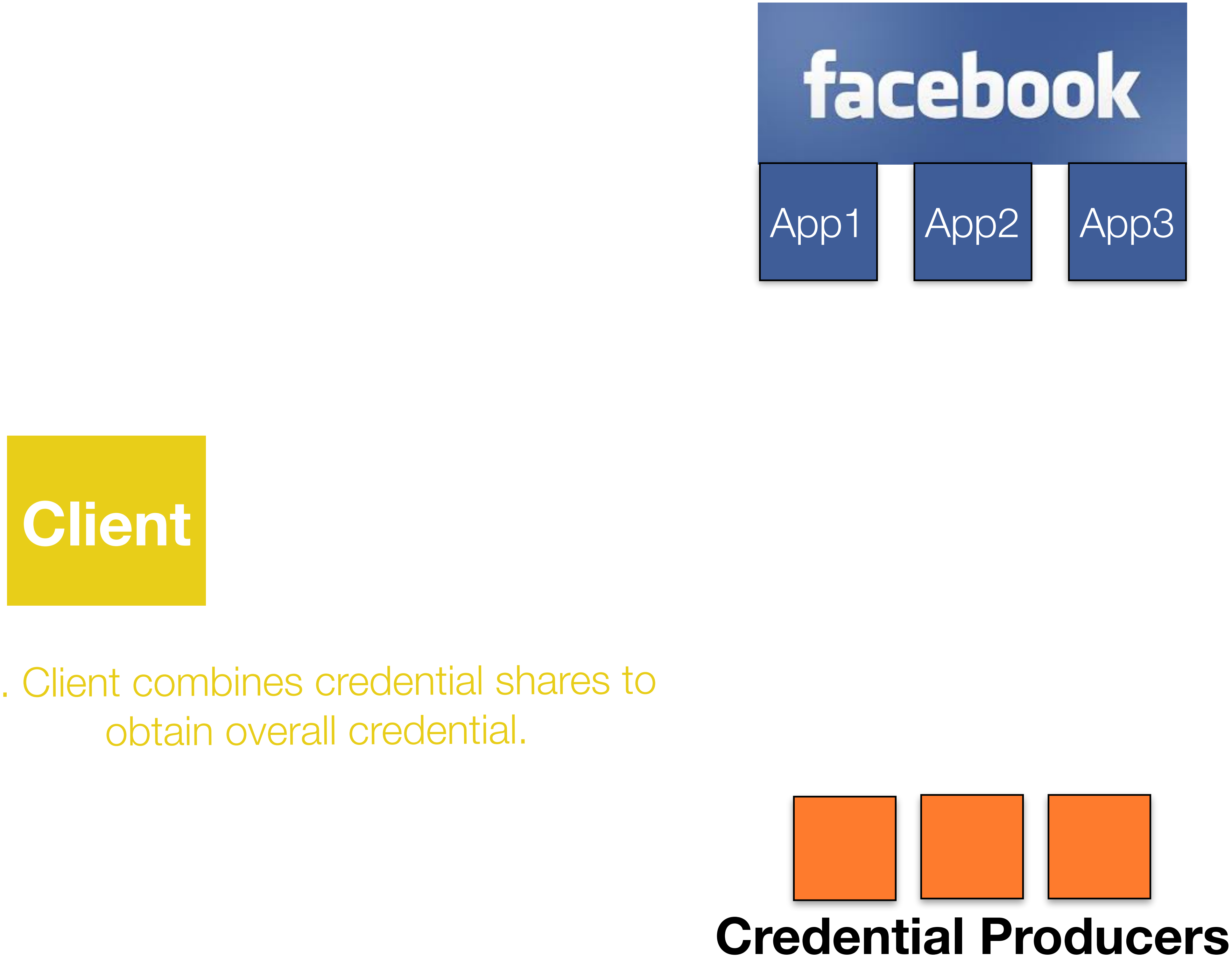
5. If OAuth token verified successfully, each credential producer returns its share of the credential to the client

Credential Producers

Credential Assignment Mechanism



Credential Assignment Mechanism



facebook

App1

App2

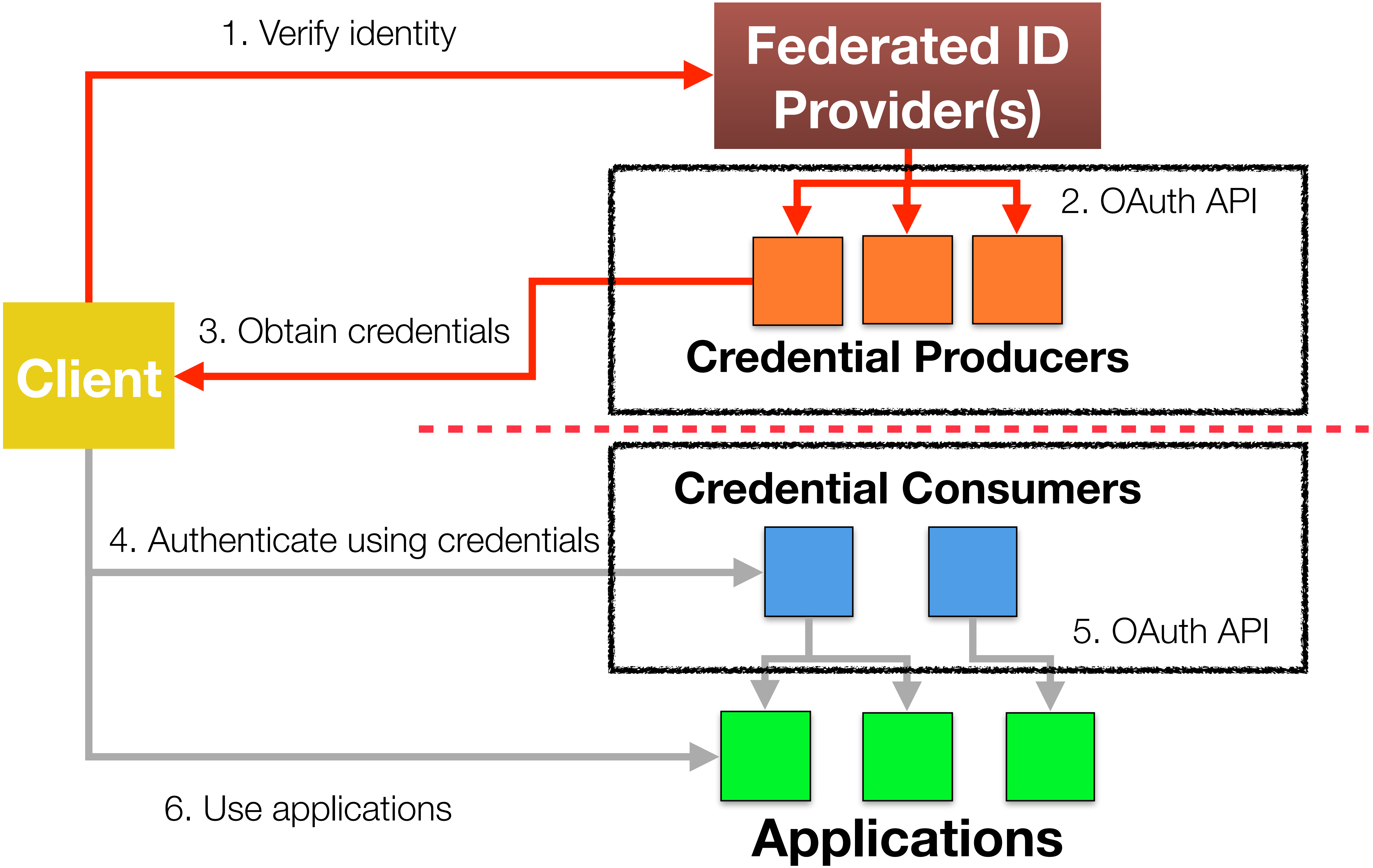
App3

Client

7. Client combines credential shares to obtain overall credential.

Credential Producers

System Architecture



1. Verify identity

Federated ID Provider(s)

2. OAuth API

Credential Producers

Client

3. Obtain credentials

Credential Consumers

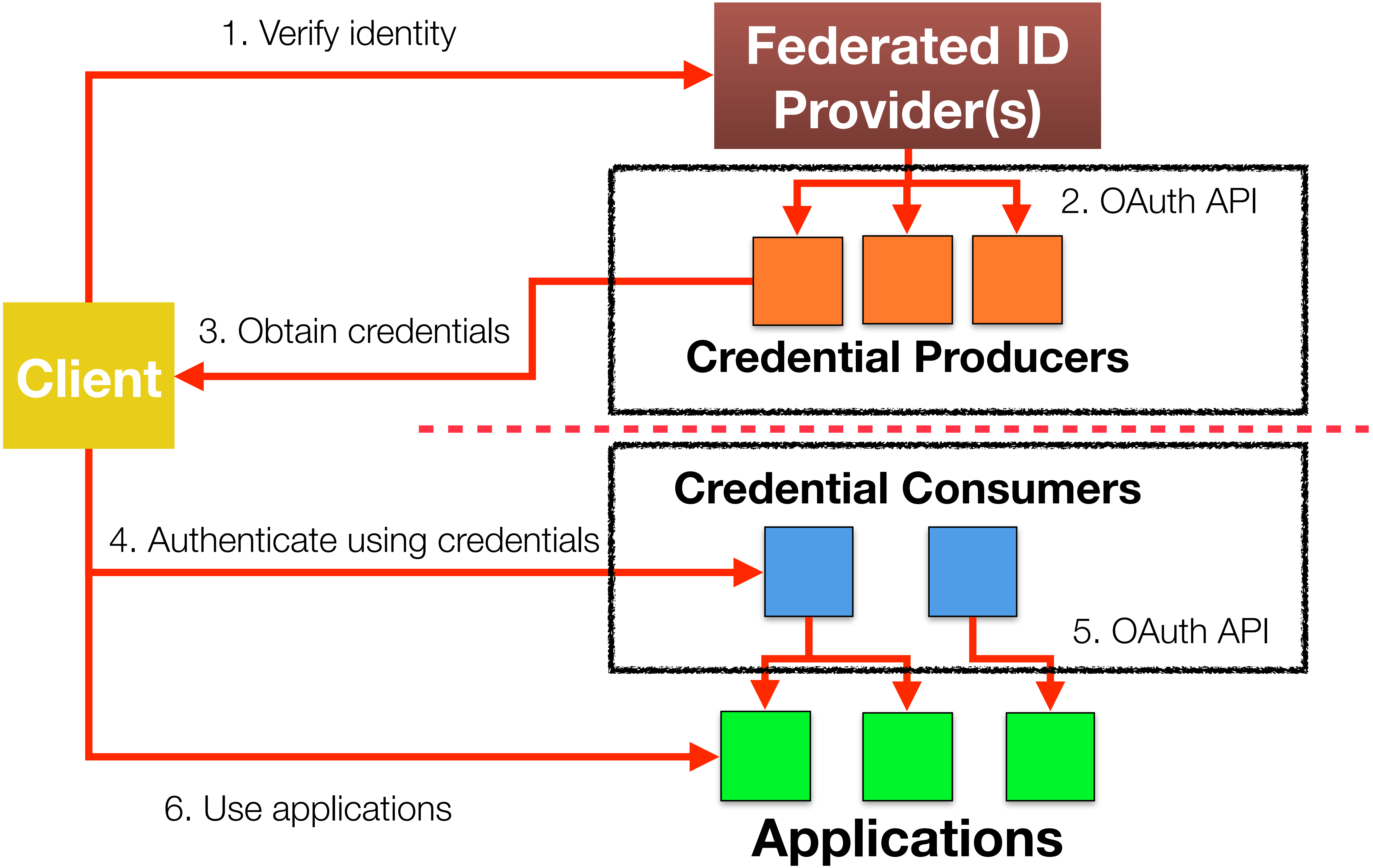
4. Authenticate using credentials

5. OAuth API

6. Use applications

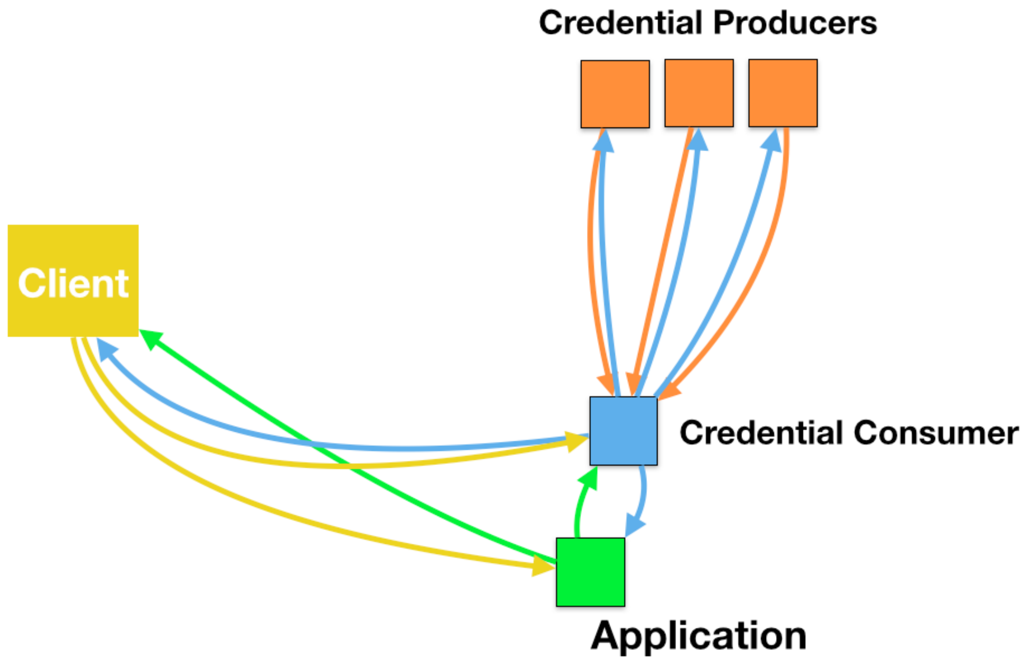
Applications

System Architecture

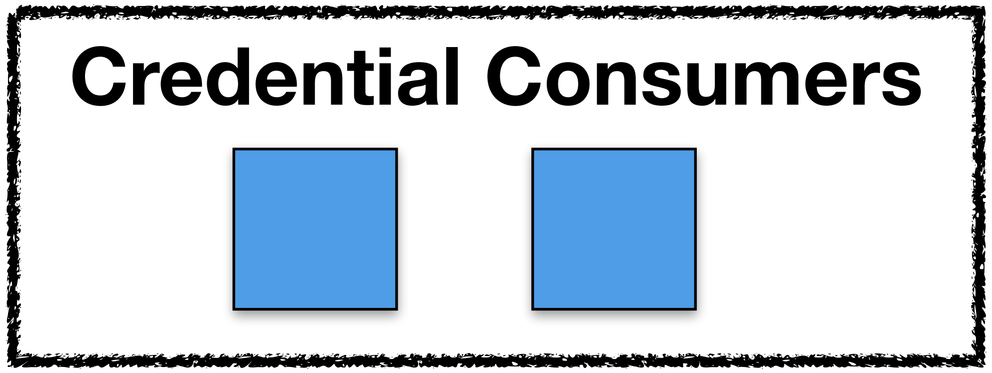


Credential Consumers

Authenticating with and using privacy preserving credentials



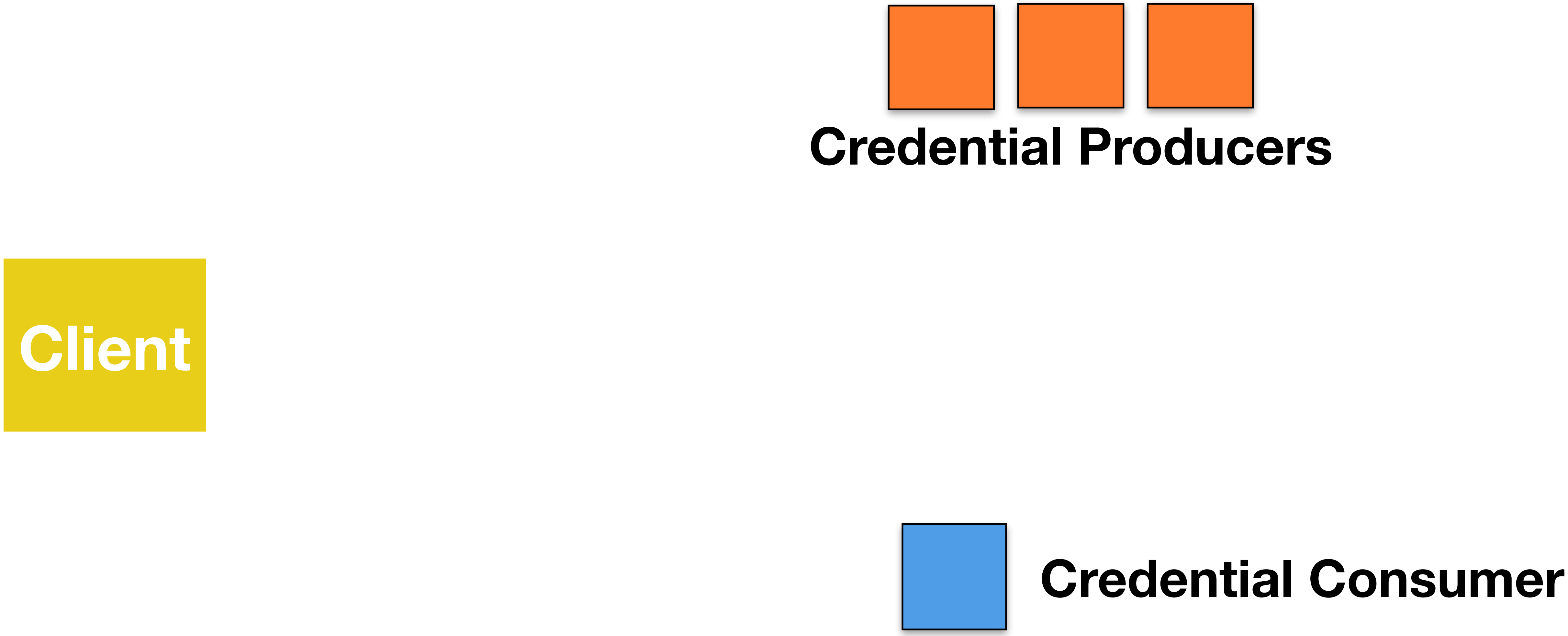
Credential Consumers



- Map credentials to pseudonyms
- Pseudonyms produced are not linkable back to federated IDs
- **OAuth provider consumers:** Expose pseudonym IDs to applications via OAuth.

  Easily integrate with applications already using federated authentication
- **Application-embedded consumer** directly in application

System Architecture

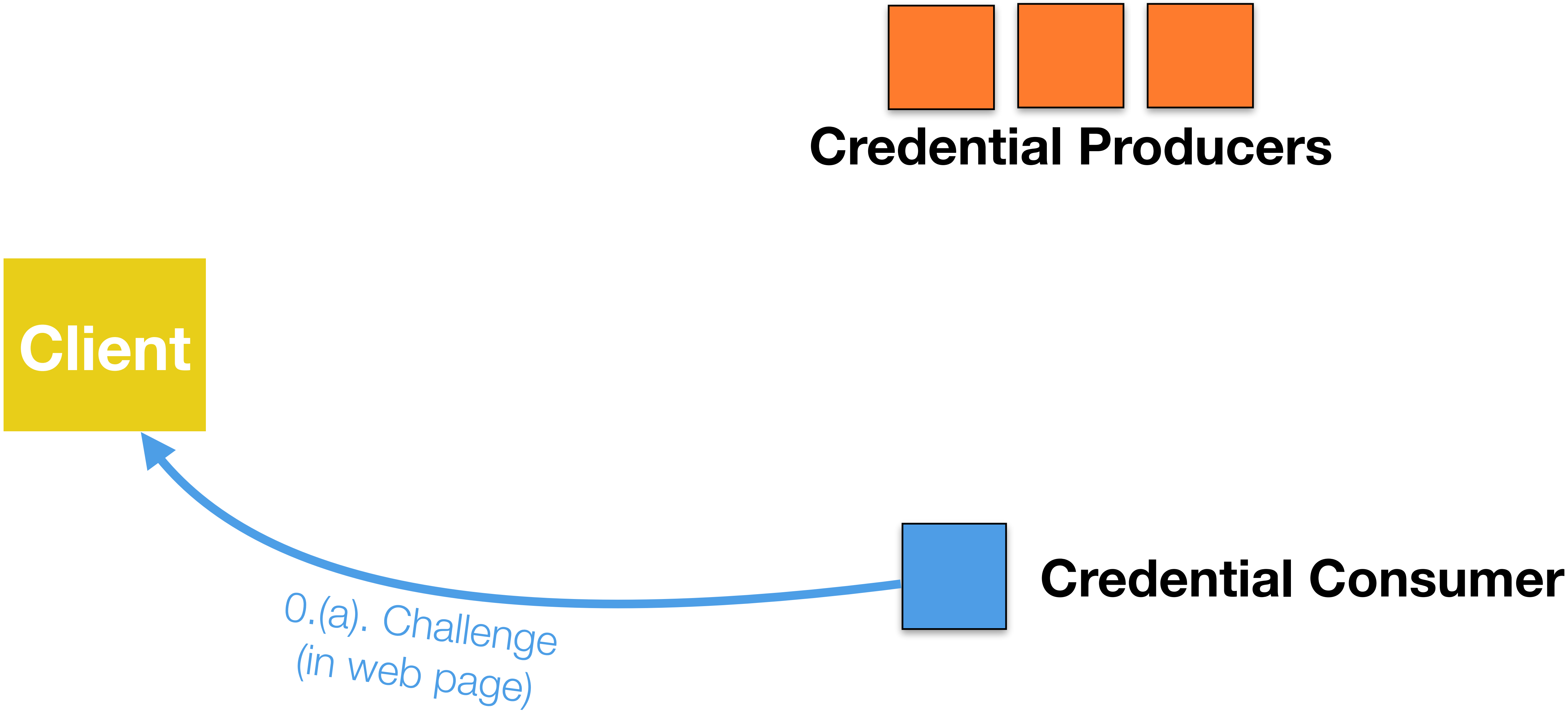


Credential Producers

Client

Credential Consumer

System Architecture



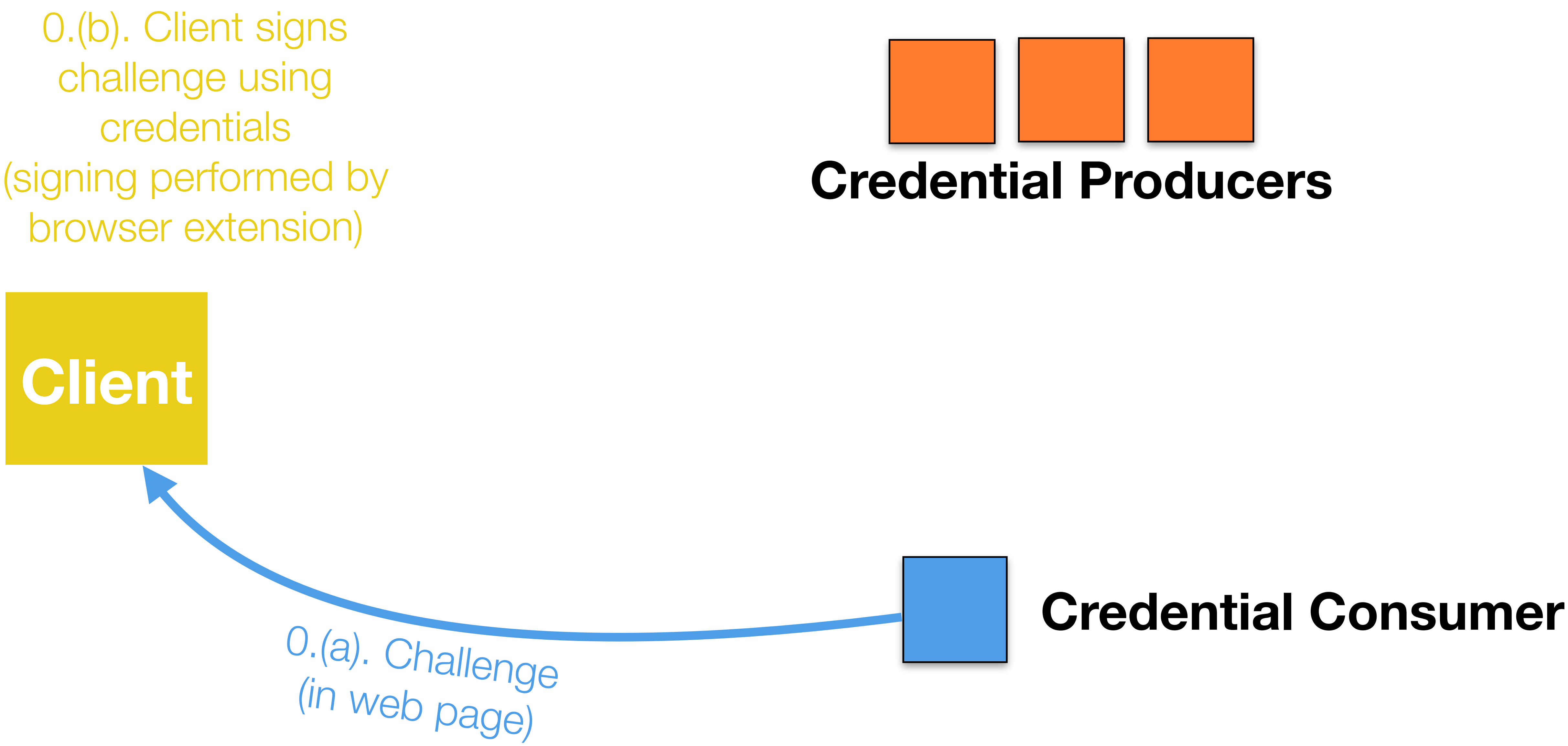
Credential Producers

Client

0.(a). Challenge (in web page)

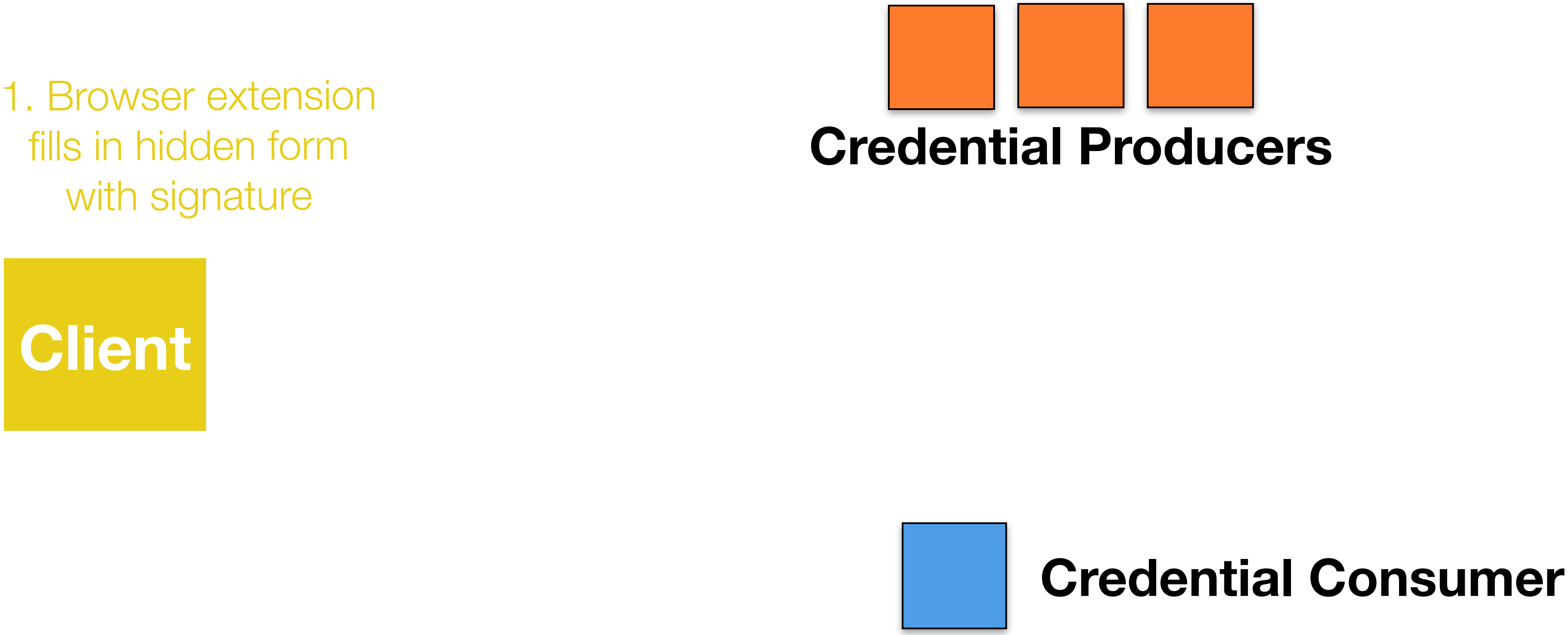
Credential Consumer

System Architecture



0.(b). Client signs challenge using credentials (signing performed by browser extension)

System Architecture



1. Browser extension fills in hidden form with signature

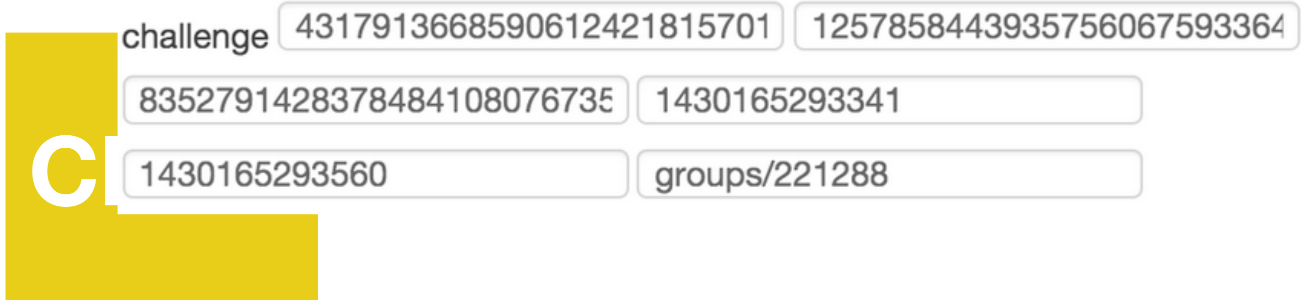
Credential Producers

Client

Credential Consumer

System Architecture

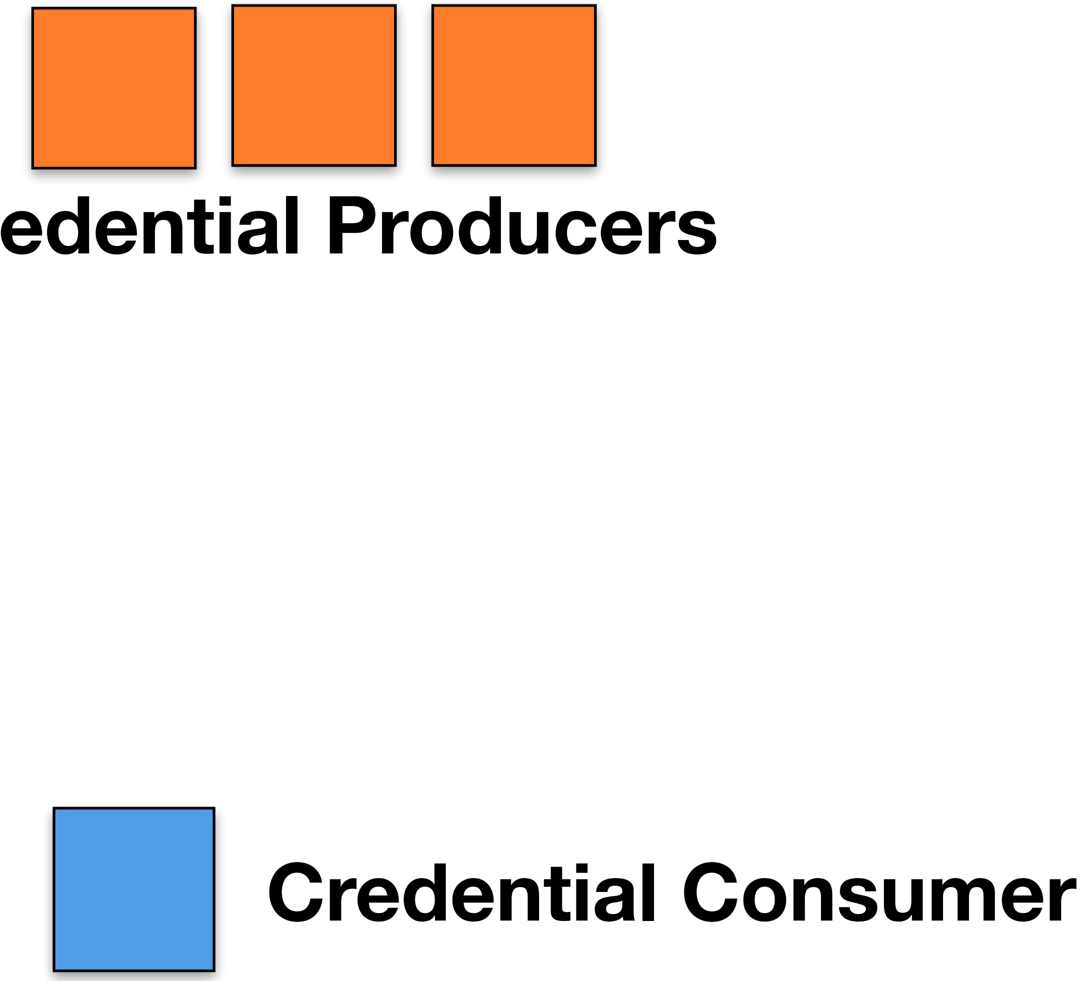
1. Browser extension fills in hidden form with signature



challenge 4317913668590612421815701 1257858443935756067593364

8352791428378484108076735 1430165293341

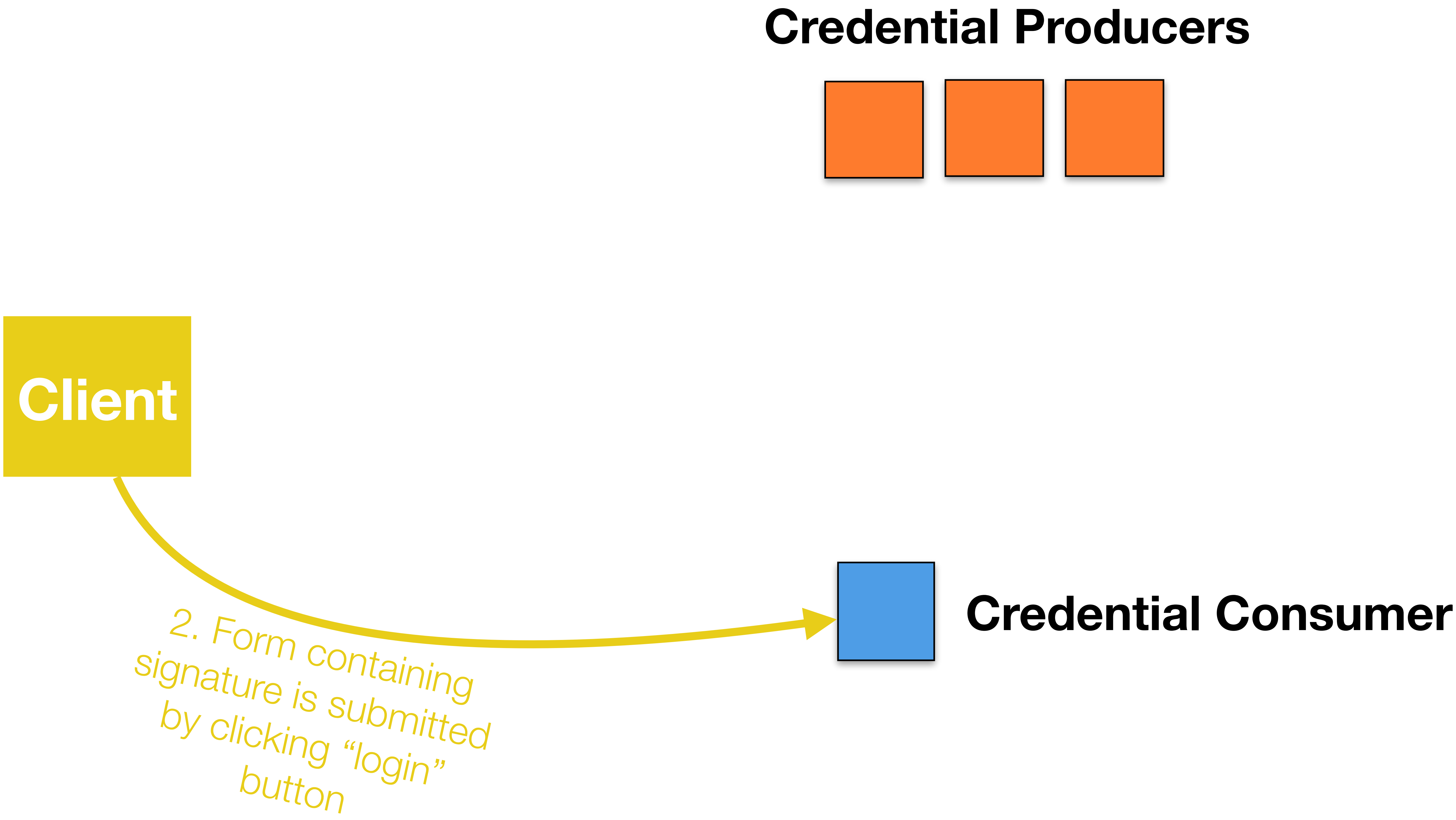
1430165293560 groups/221288



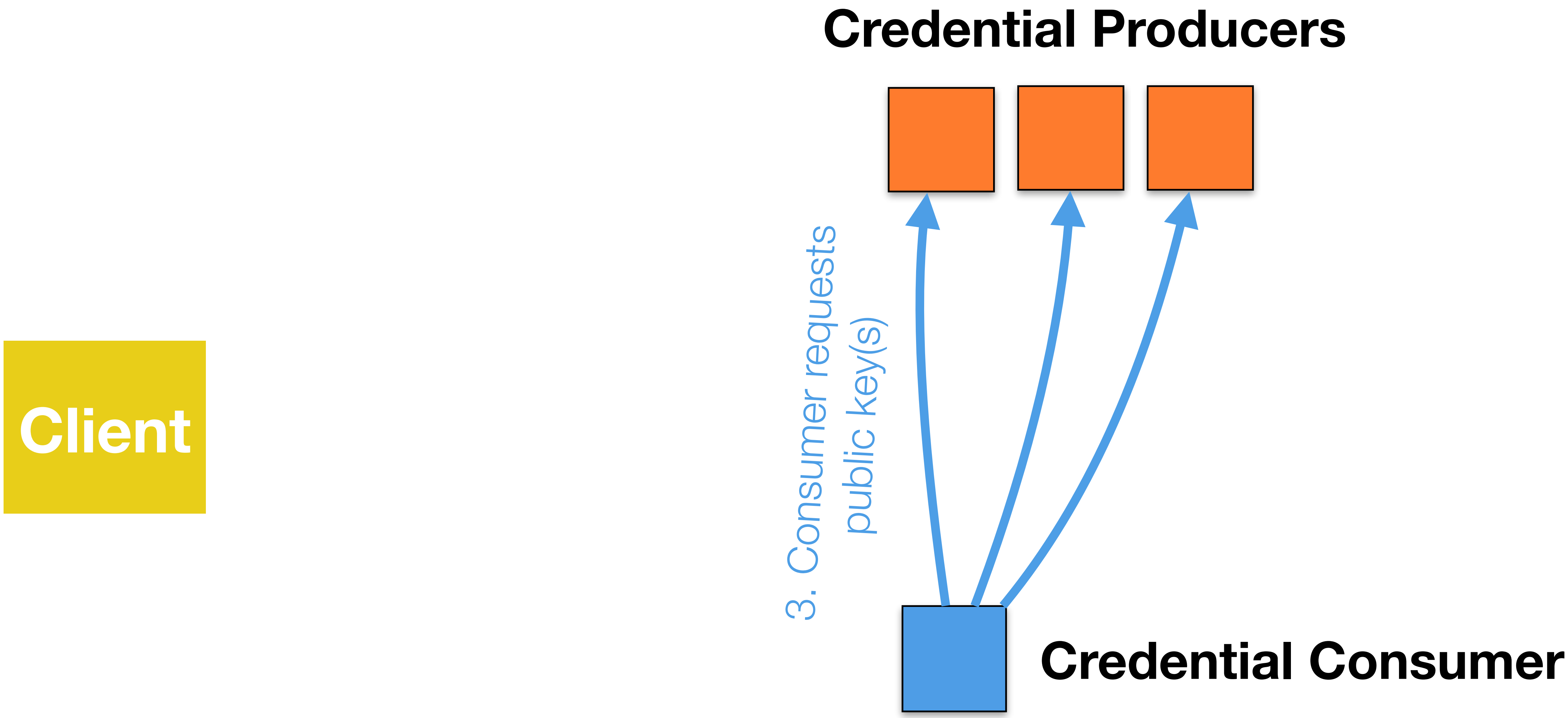
Credential Producers

Credential Consumer

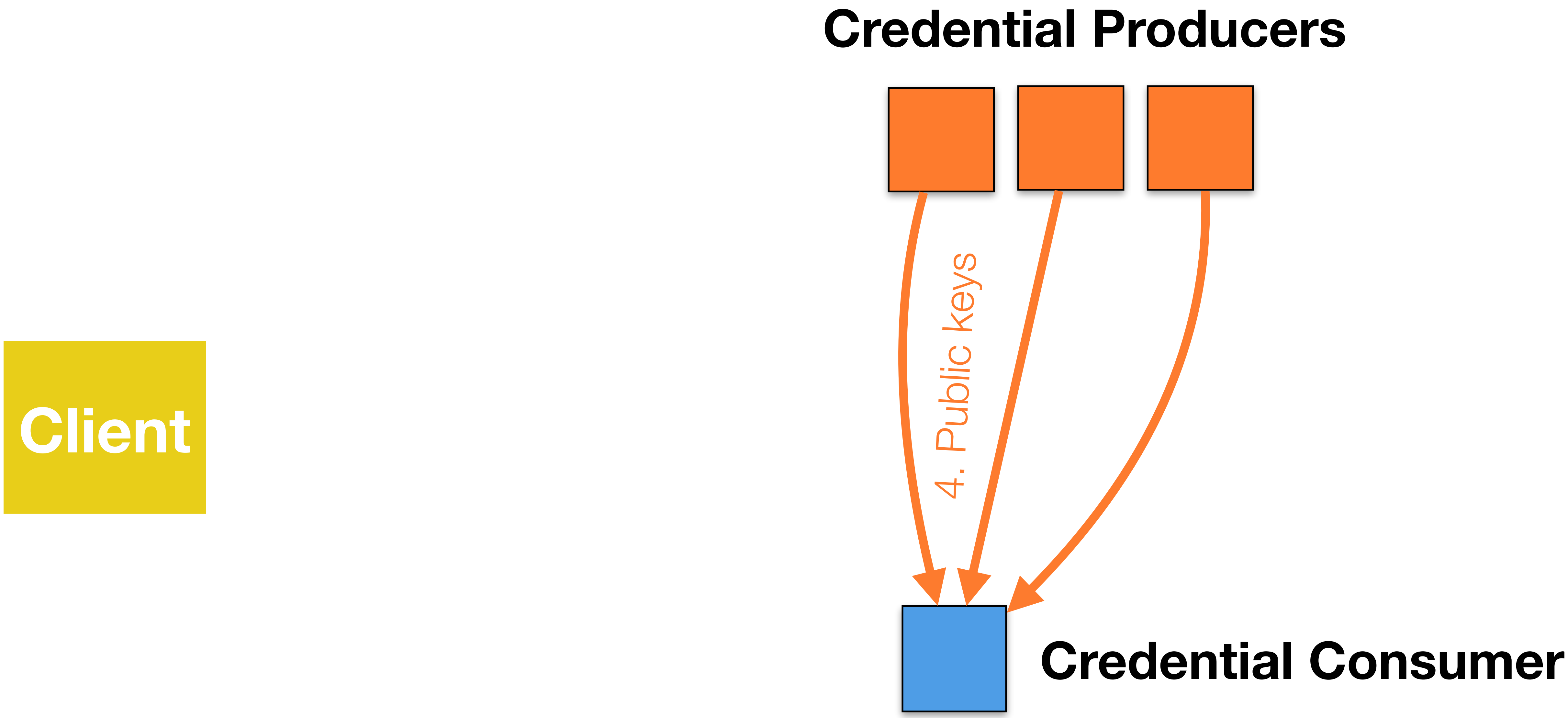
System Architecture


Credential Producers
Client
2. Form containing signature is submitted by clicking "login" button
Credential Consumer

System Architecture



System Architecture



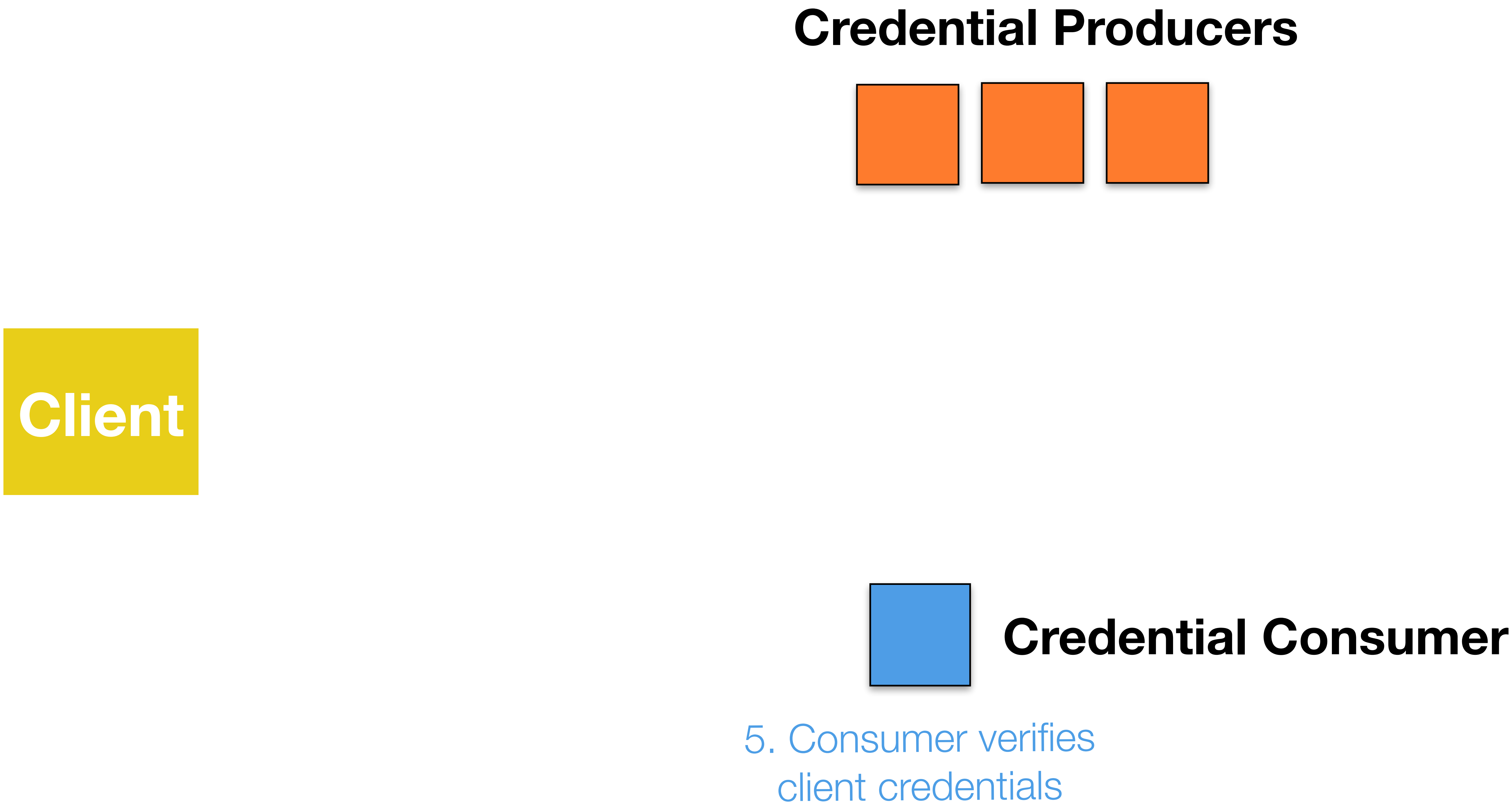
Credential Producers

4. Public keys

Client

Credential Consumer

System Architecture



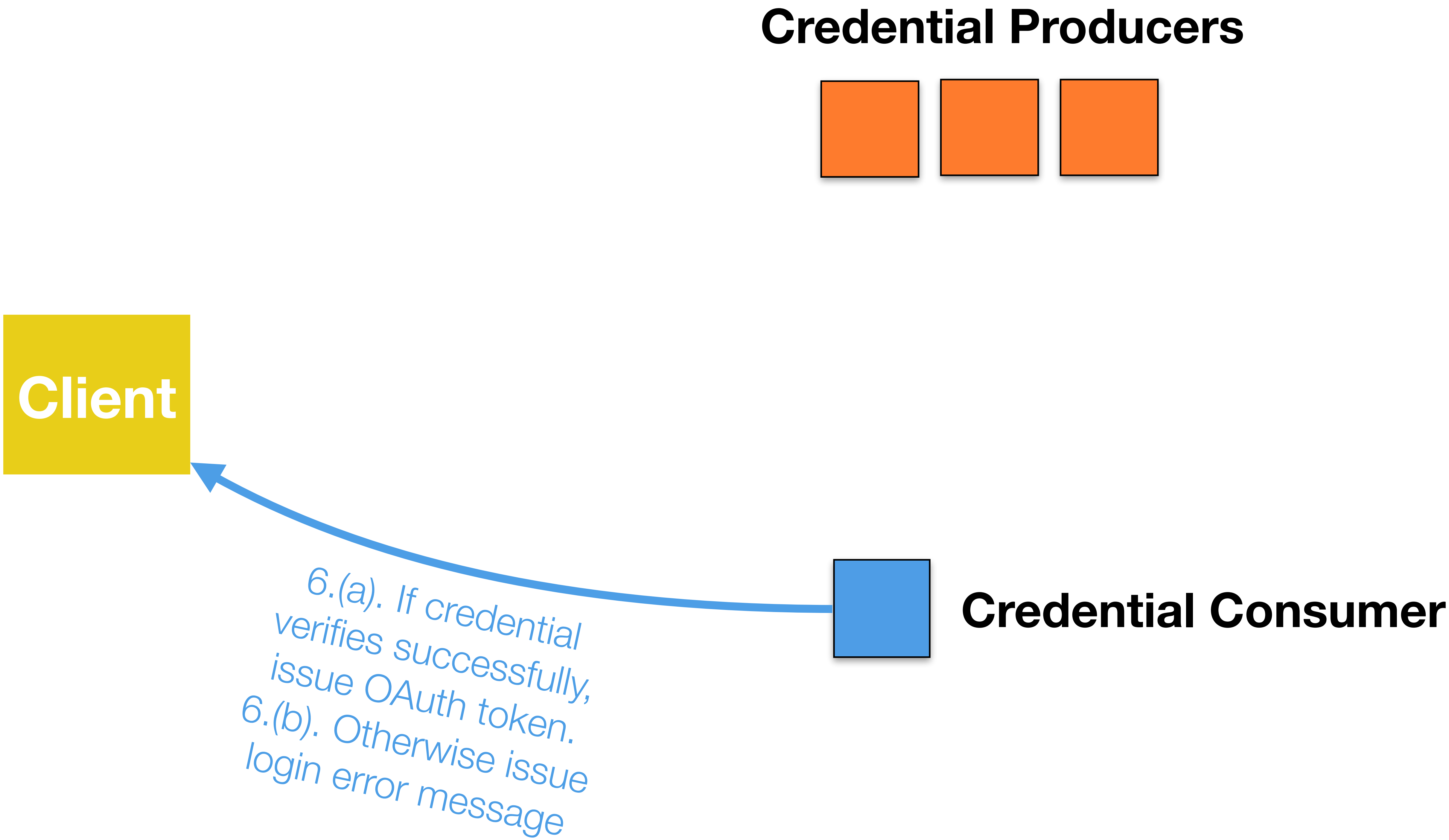
Credential Producers

Client

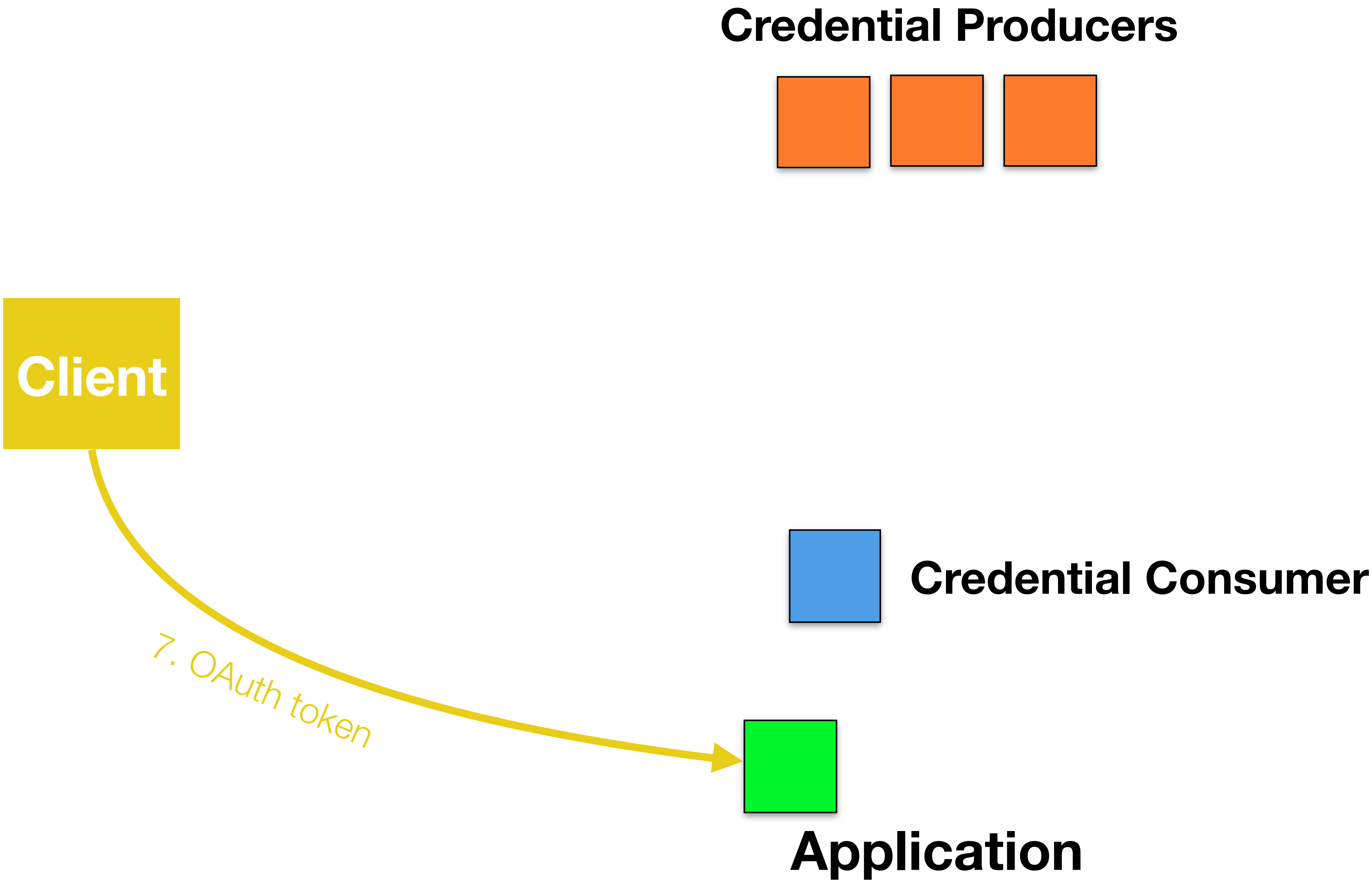
Credential Consumer

5. Consumer verifies client credentials

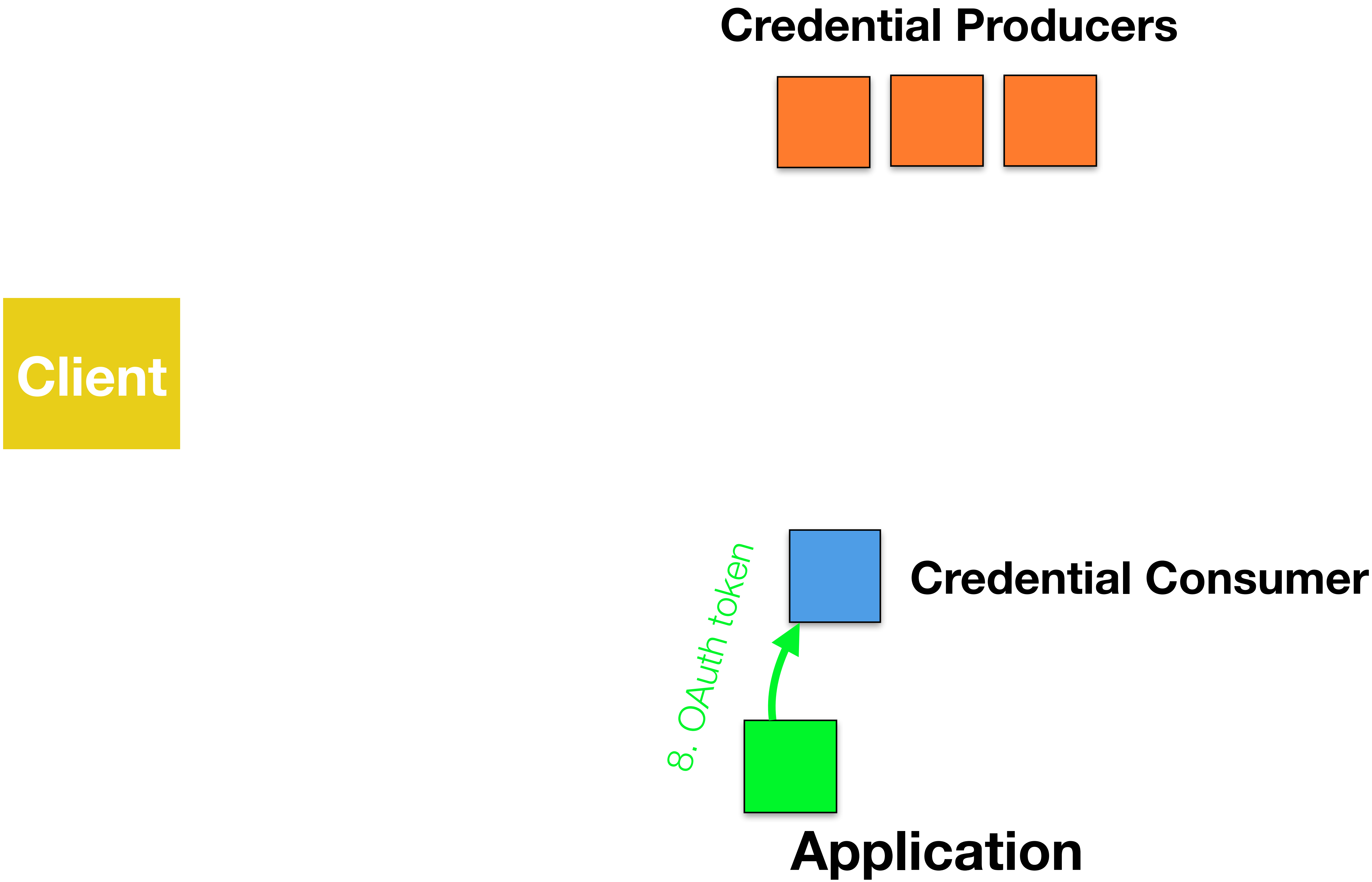
System Architecture


Credential Producers
Client
6.(a). If credential verifies successfully, issue OAuth token.
6.(b). Otherwise issue login error message
Credential Consumer

System Architecture



System Architecture



Credential Producers

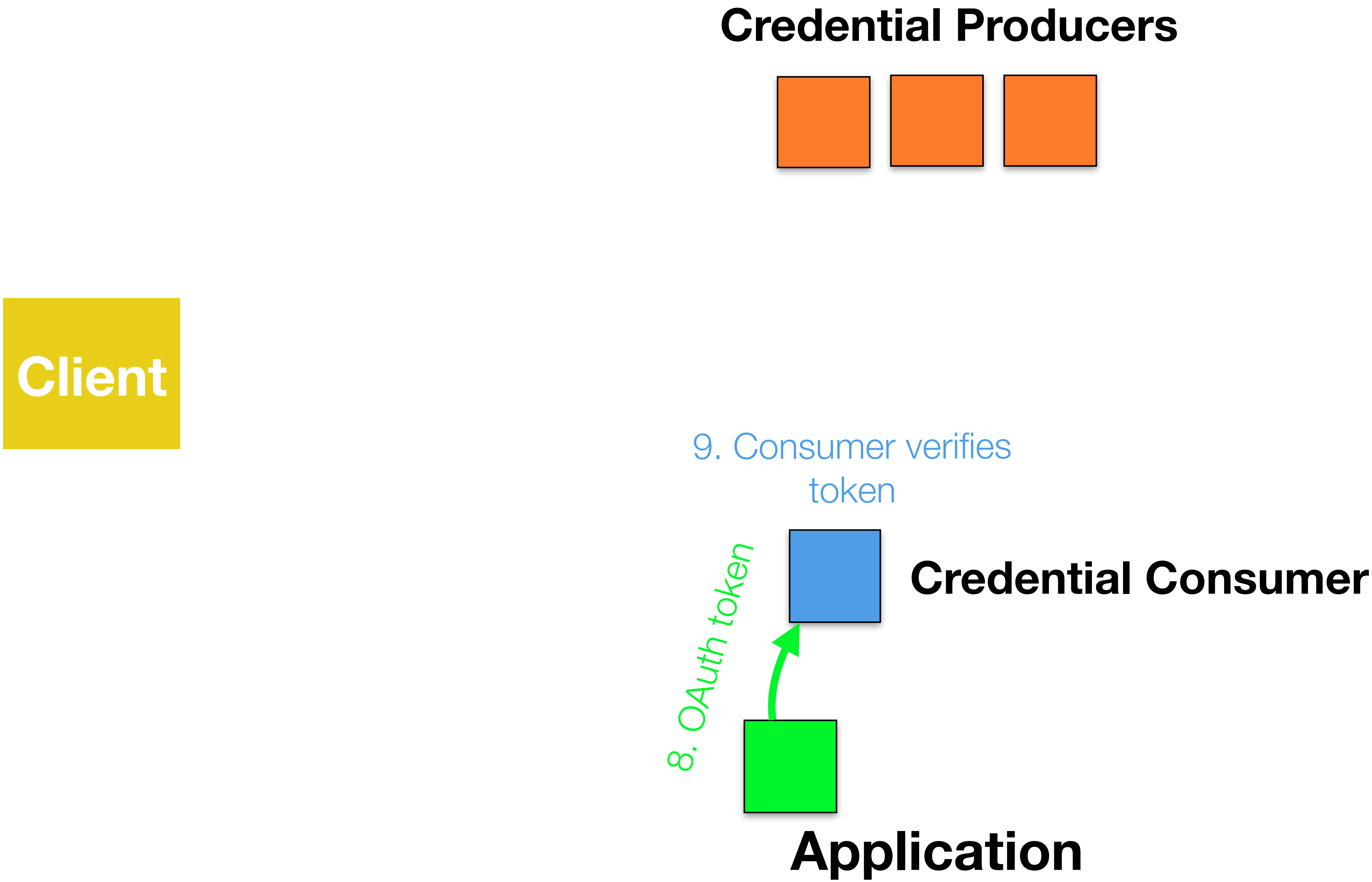
Client

8. OAuth token

Credential Consumer

Application

System Architecture



Credential Producers

Client

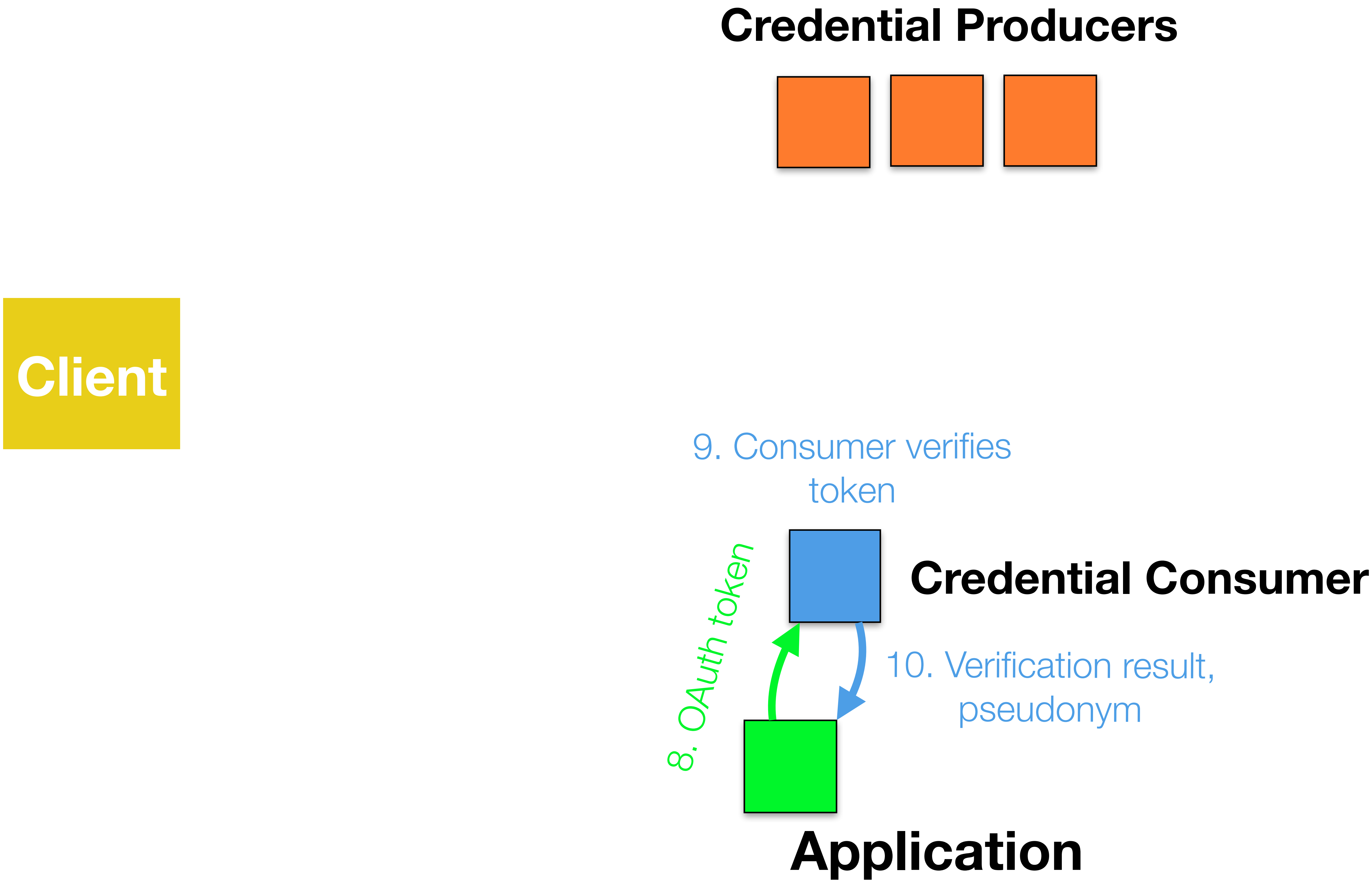
9. Consumer verifies token

8. OAuth token

Credential Consumer

Application

System Architecture



Credential Producers

Client

9. Consumer verifies token

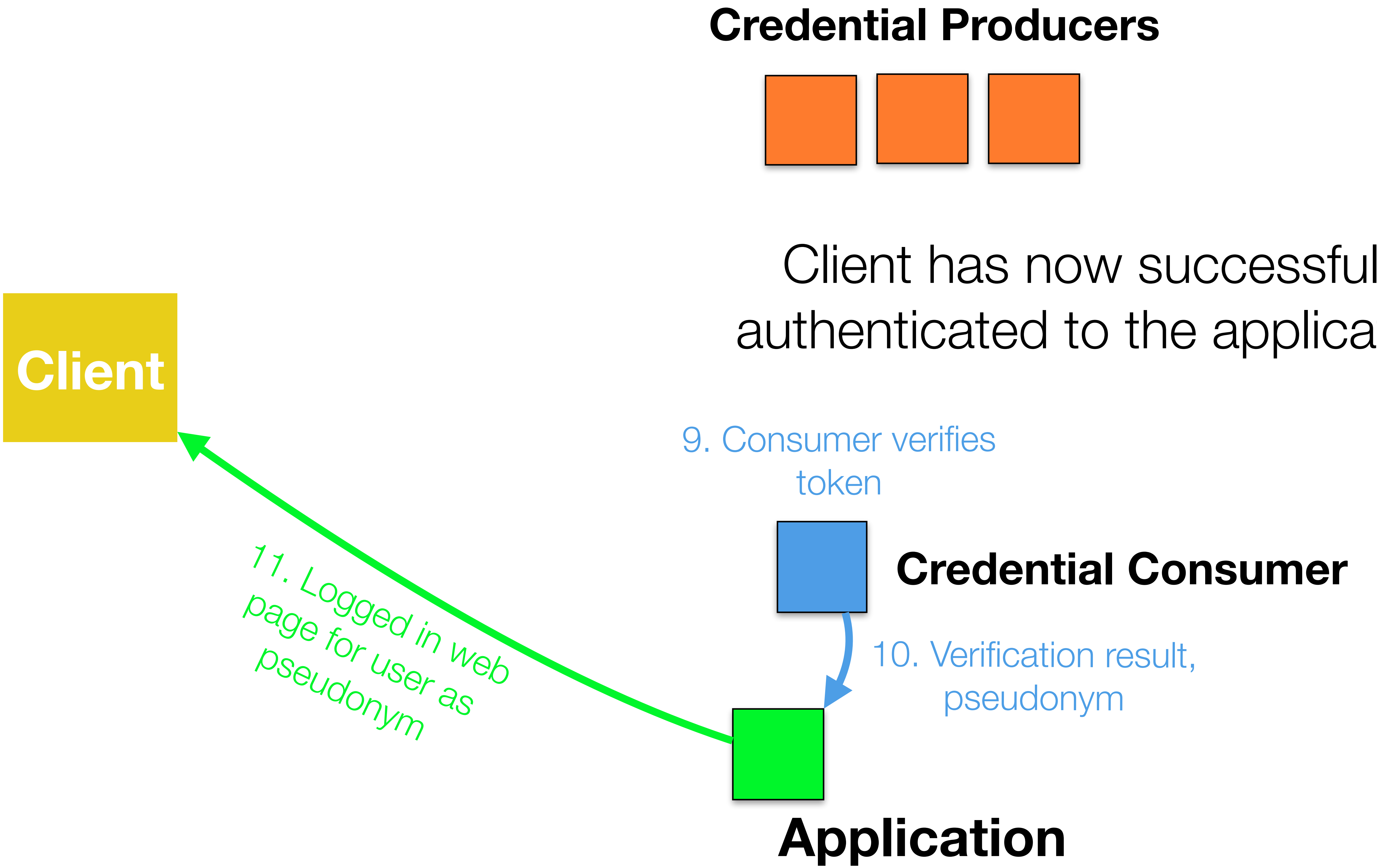
8. OAuth token

Credential Consumer

10. Verification result, pseudonym

Application

System Architecture



Credential Producers

Client has now successfully authenticated to the application

Client

9. Consumer verifies token

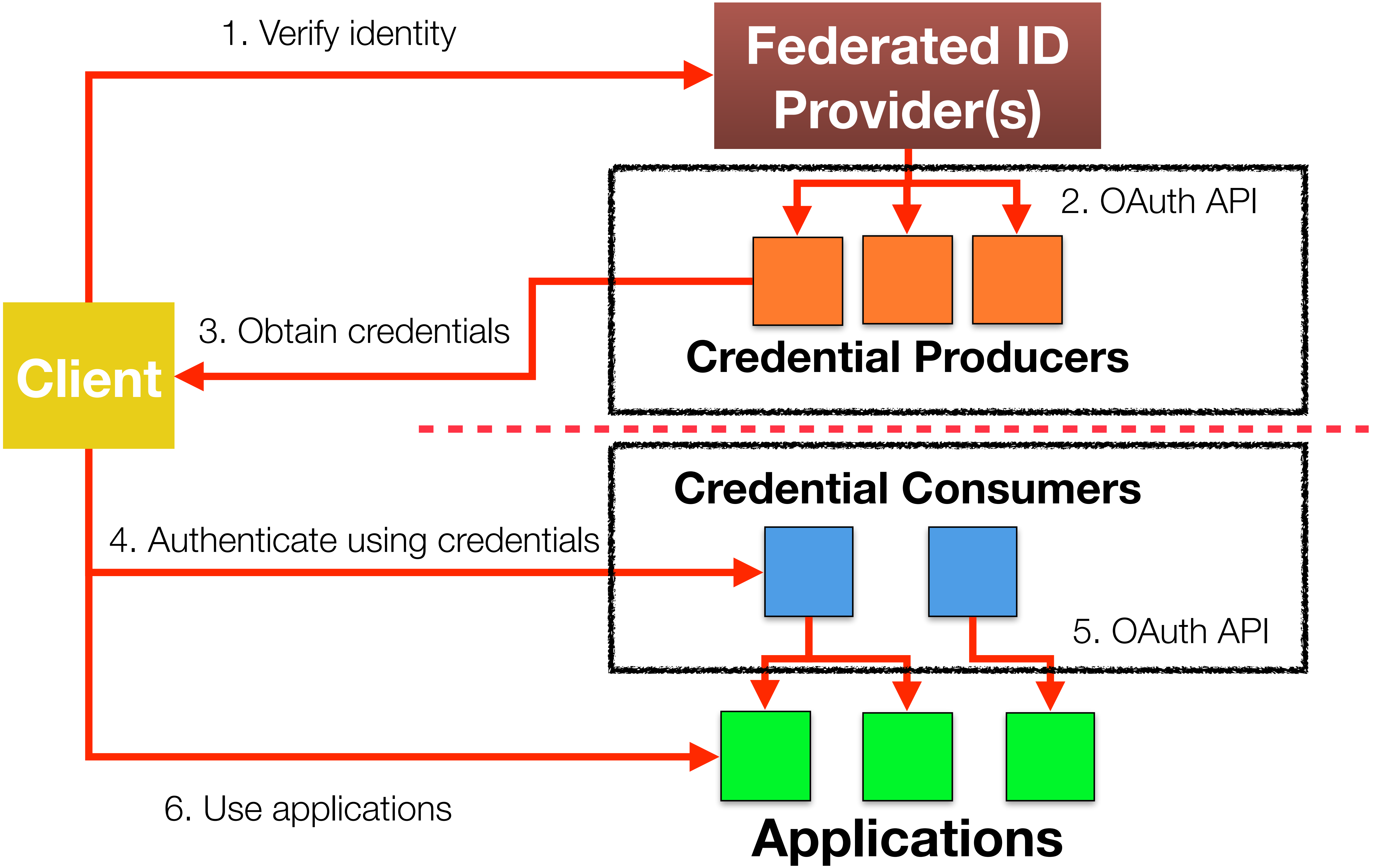
Credential Consumer

10. Verification result, pseudonym

11. Logged in web page for user as pseudonym

Application

System Architecture

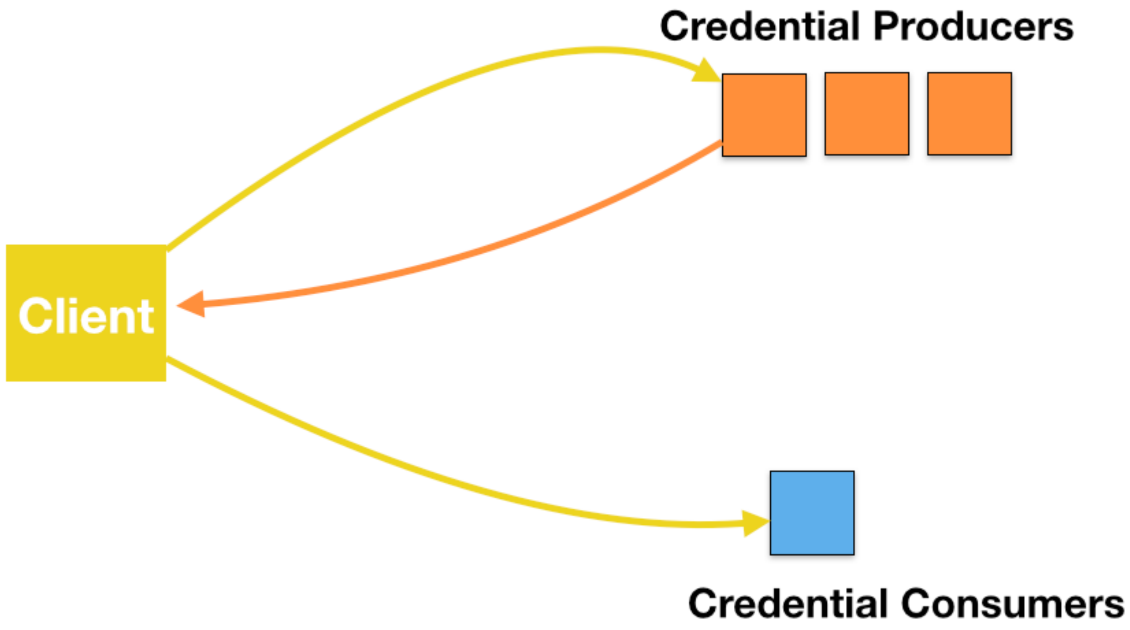


Roadmap

1. Background
2. Work Overview
3. System Architecture
4. Credential Producers and Consumers
   - **At -Large Credentials**
   - Group Credentials
5. Evaluation
6. Conclusions

At-Large Credential Scheme

Can use for privacy preserving Wikipedia login



At-Large Credential Scheme

- Represents that the user has been verified as the owner of *some* federated identity.
- Anonymity set is implicitly the users who have collected a credential
- Accountability through rate limiting: producers restrict number of credentials a federated ID gets within a period of time
- Can include credential attributes, such as “age over 18” or “identity active for at least one year”

Technical Building Block: Blind Signatures

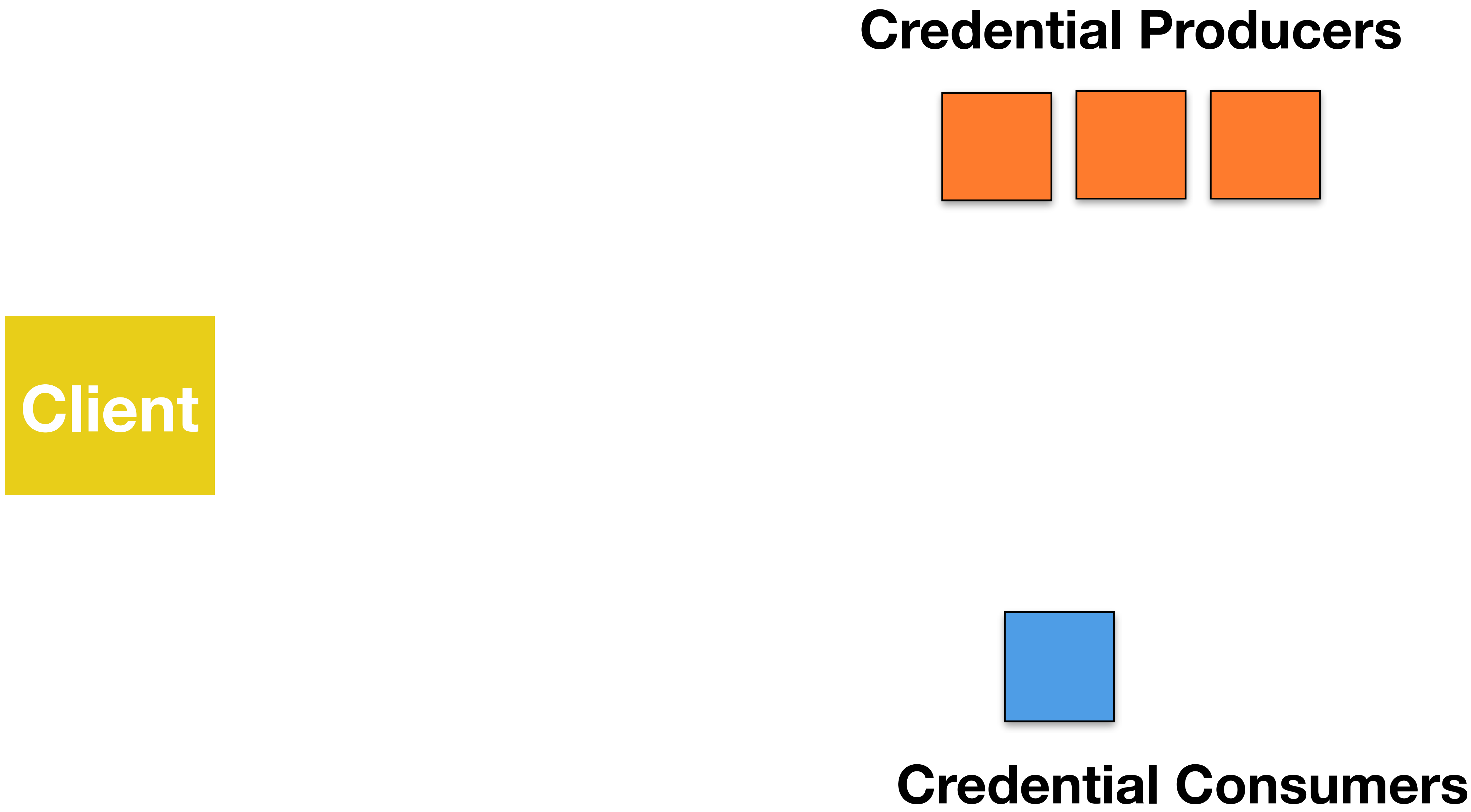
1. Request a signature on a blinded message
2. Signer cannot learn message content
3. Third party can verify unblinded signature

m —> m’ —> m’,s’ —> m,s

Technical Building Block: Blind Signatures

- Client is the requester
- Each credential producer is a signer
- Credential consumers are verifiers

At-Large Credential Scheme

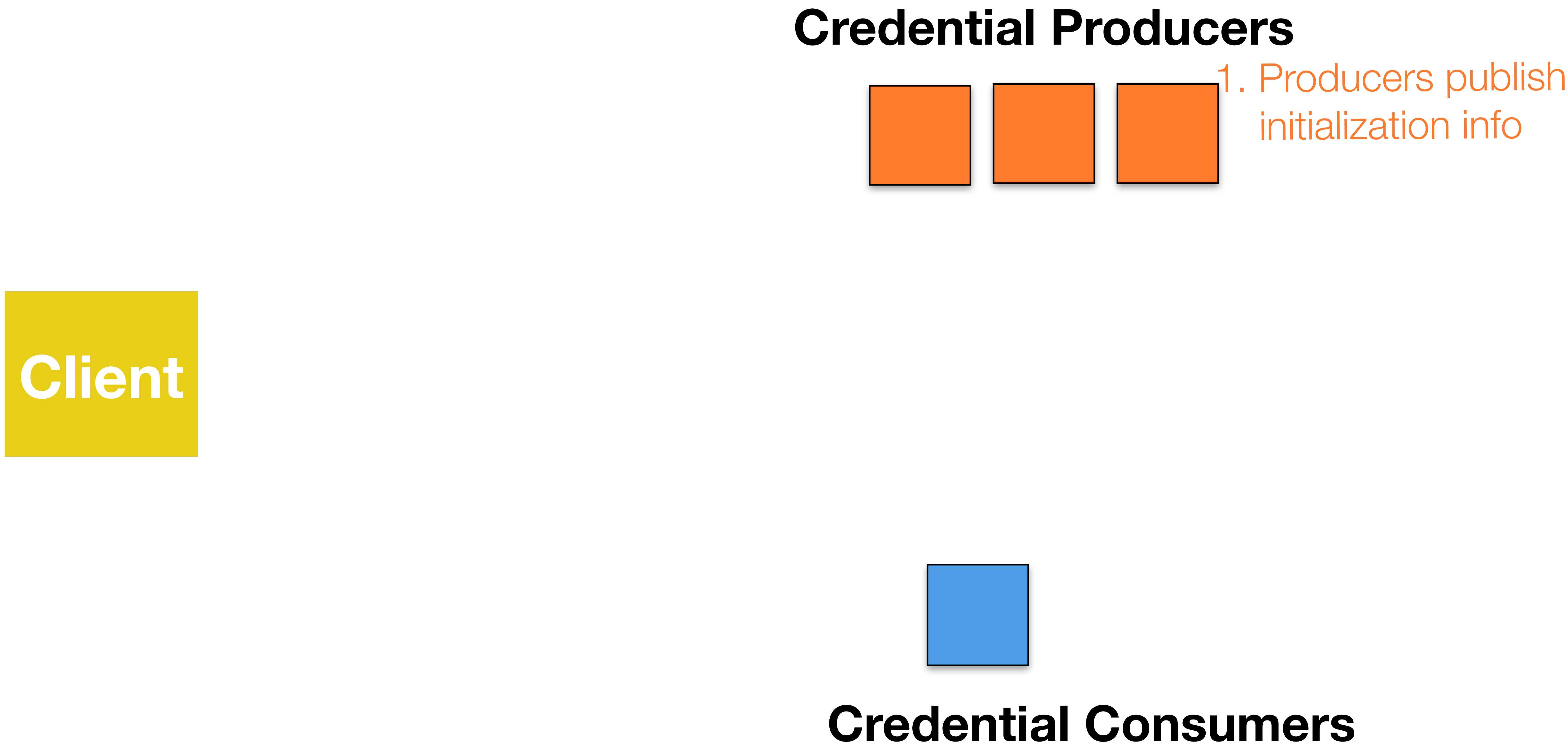


Credential Producers

Client

Credential Consumers

At-Large Credential Scheme



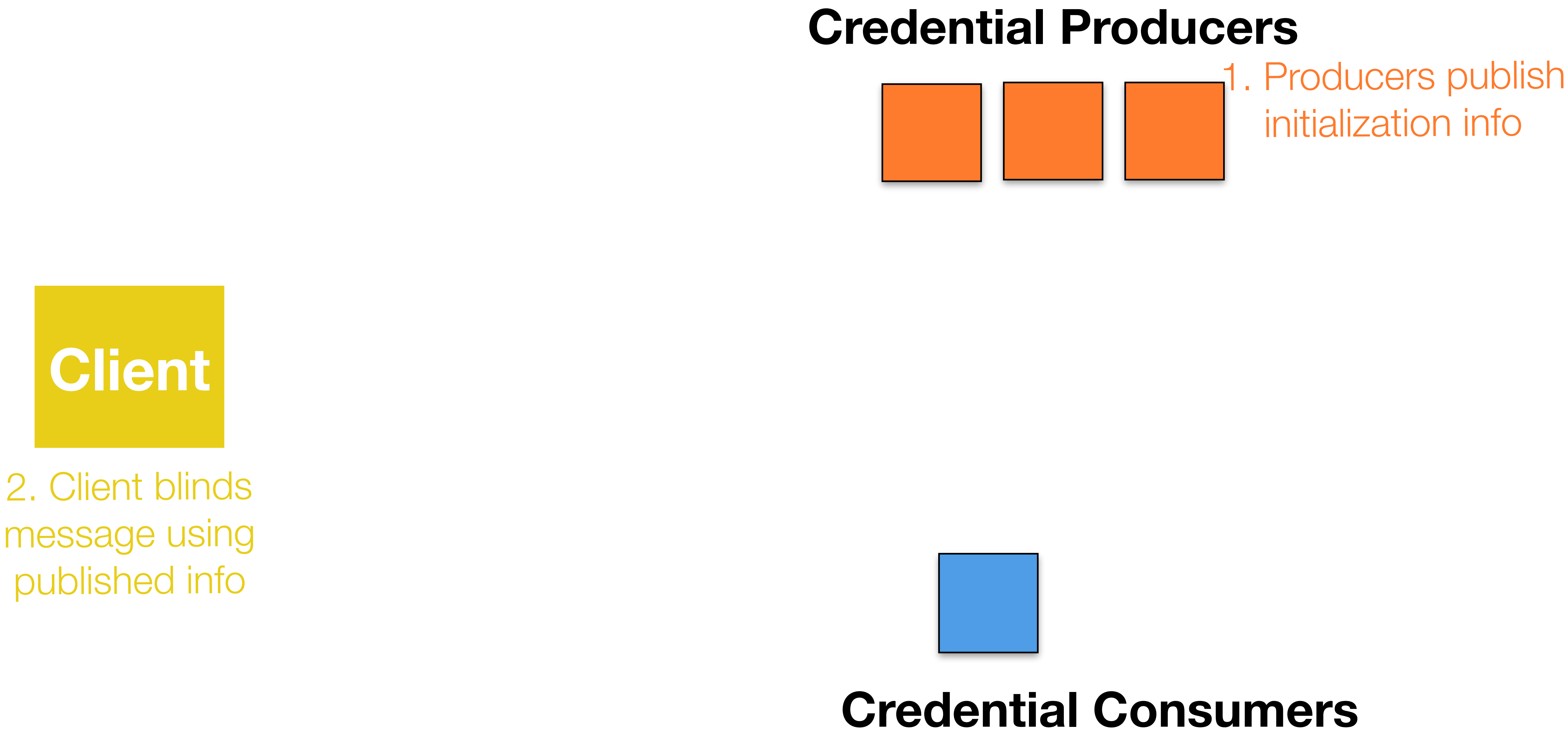
Credential Producers

1. Producers publish initialization info

Client

Credential Consumers

At-Large Credential Scheme



Credential Producers

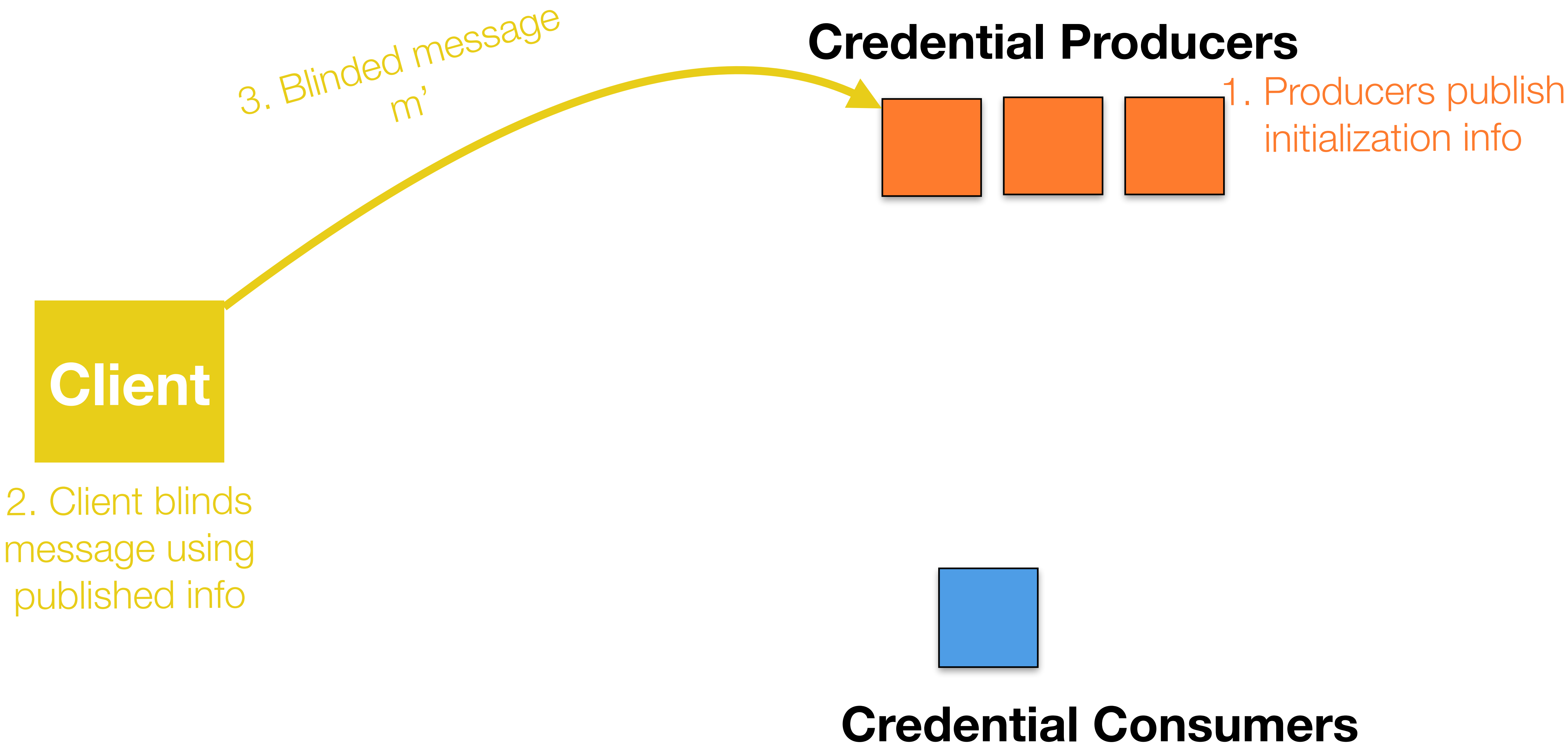
1. Producers publish initialization info

Client

2. Client blinds message using published info

Credential Consumers

At-Large Credential Scheme



Credential Producers

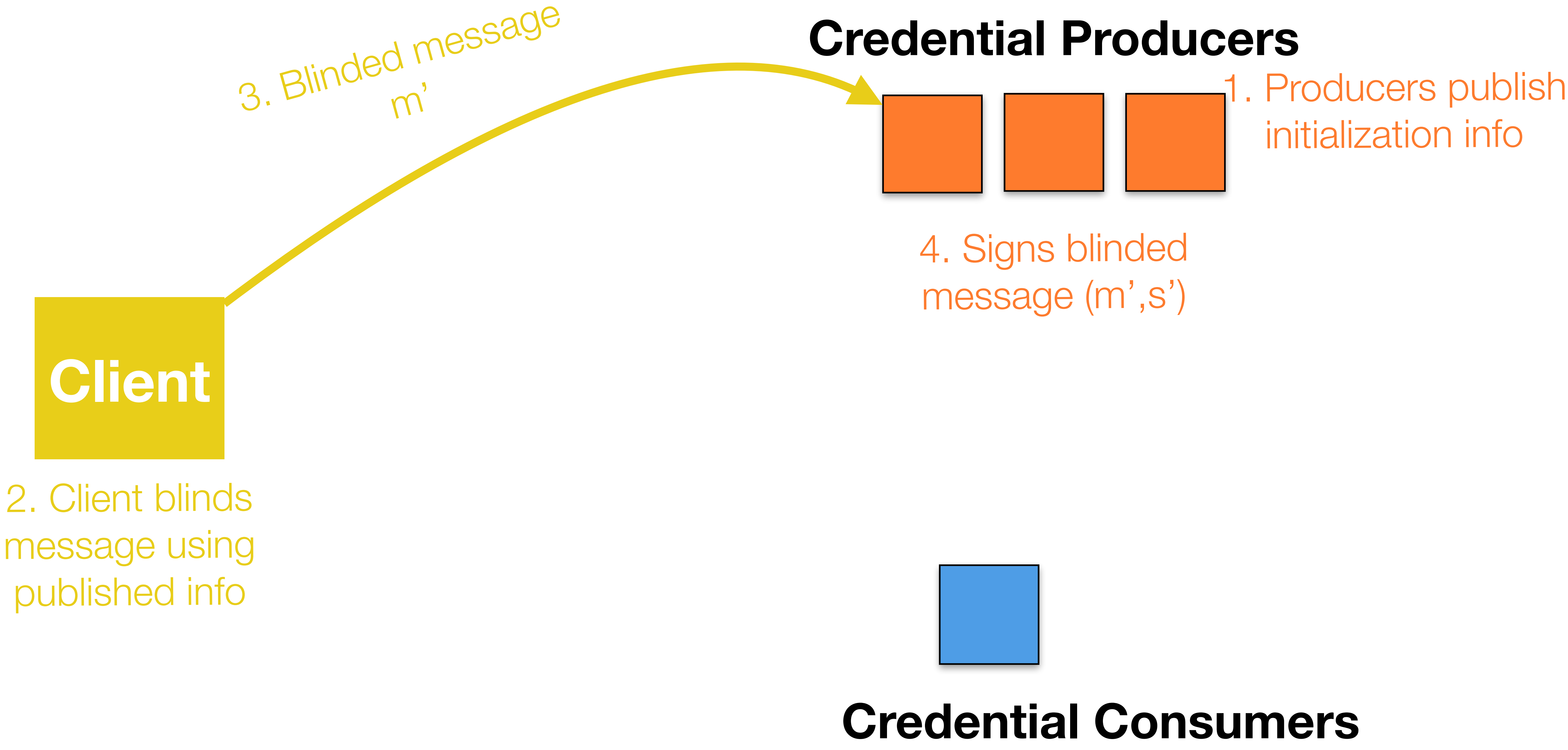
1. Producers publish initialization info

2. Client blinds message using published info

3. Blinded message m’

Credential Consumers

At-Large Credential Scheme



Credential Producers

1. Producers publish initialization info

2. Client blinds message using published info

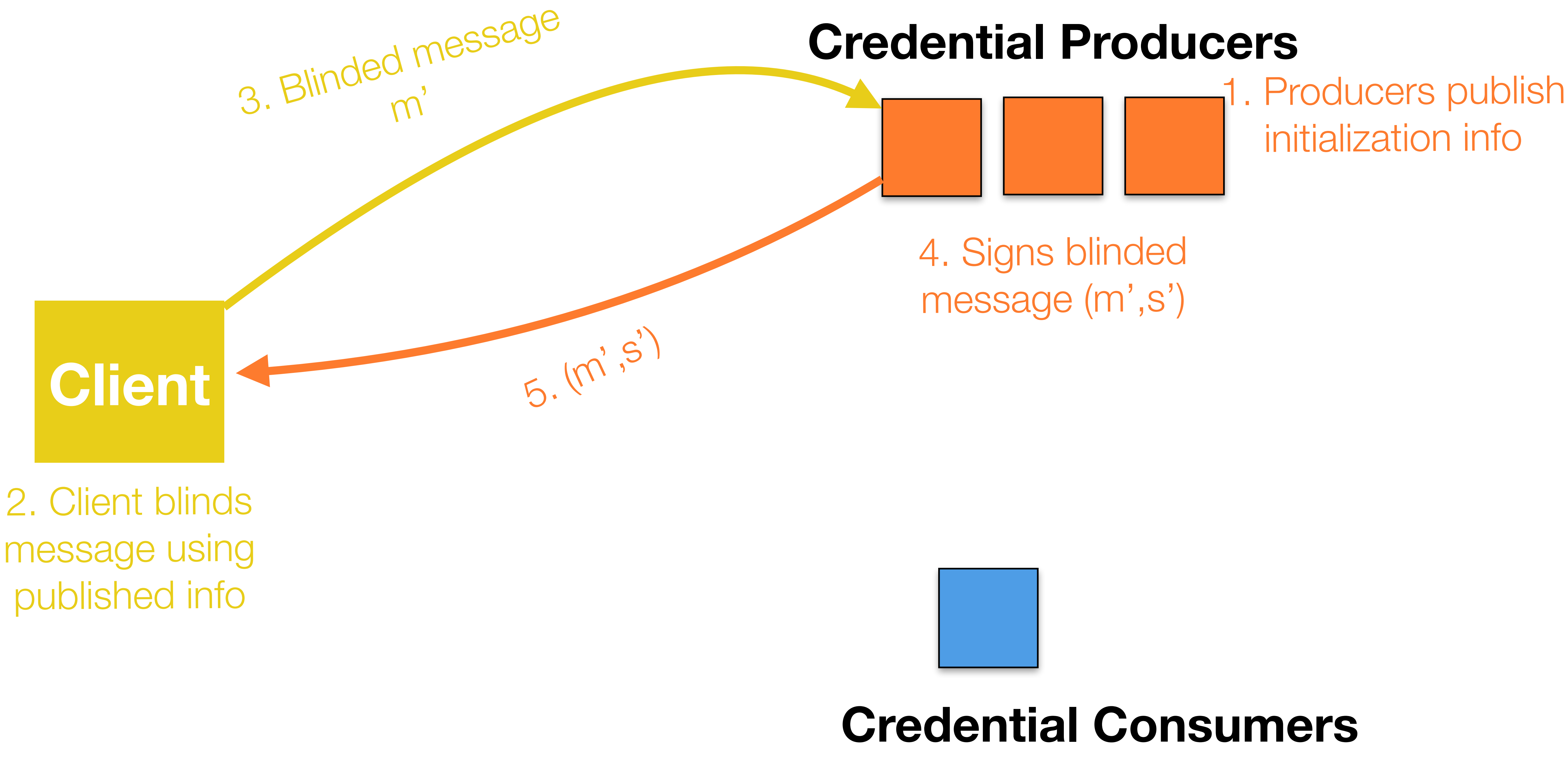
3. Blinded message m’

4. Signs blinded message (m’,s’)

Client

Credential Consumers

At-Large Credential Scheme



Credential Producers

1. Producers publish initialization info

2. Client blinds message using published info

3. Blinded message m’

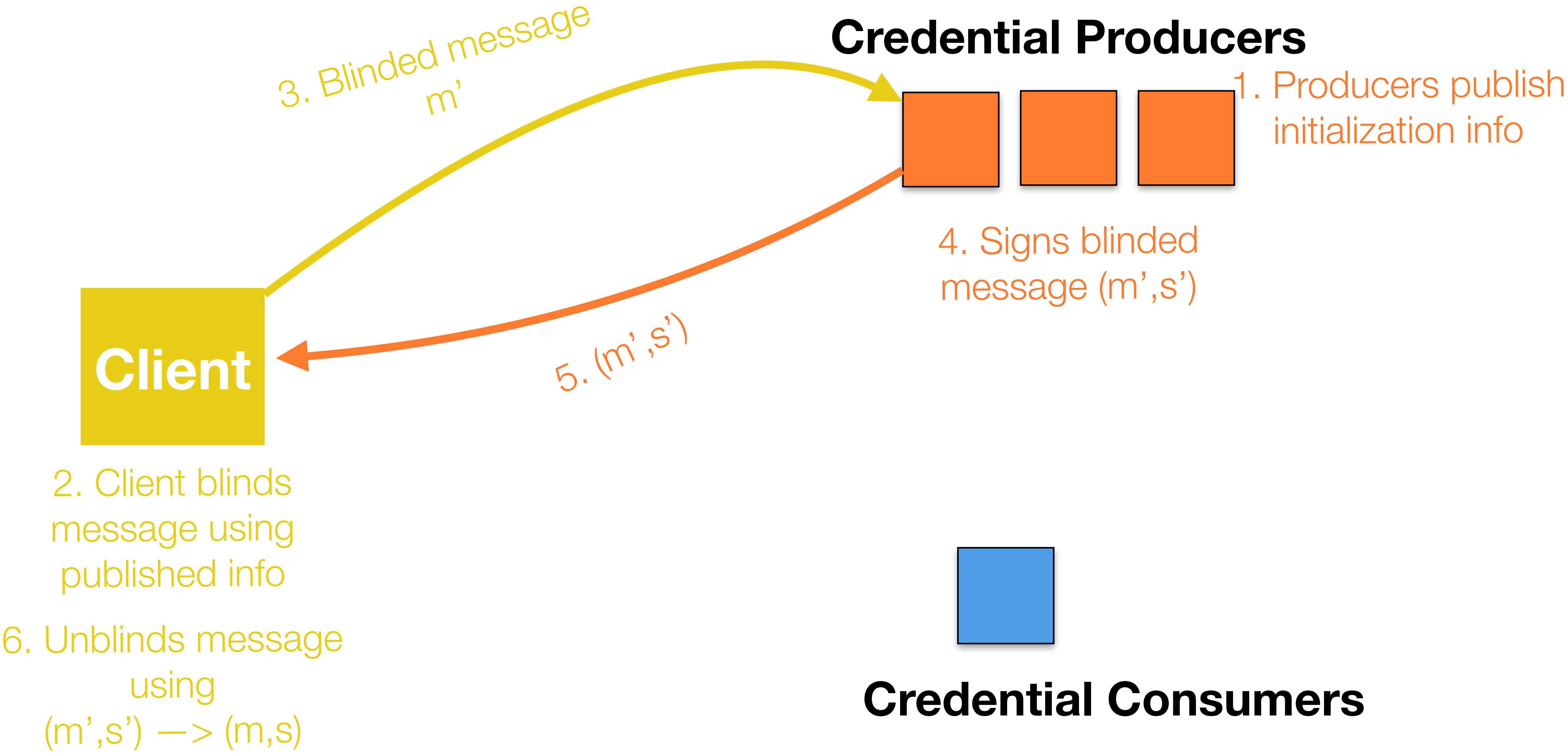
4. Signs blinded message (m’,s’)

5. (m’,s’)

Client

Credential Consumers

At-Large Credential Scheme



Credential Producers

1. Producers publish initialization info

Client

2. Client blinds message using published info

3. Blinded message m’

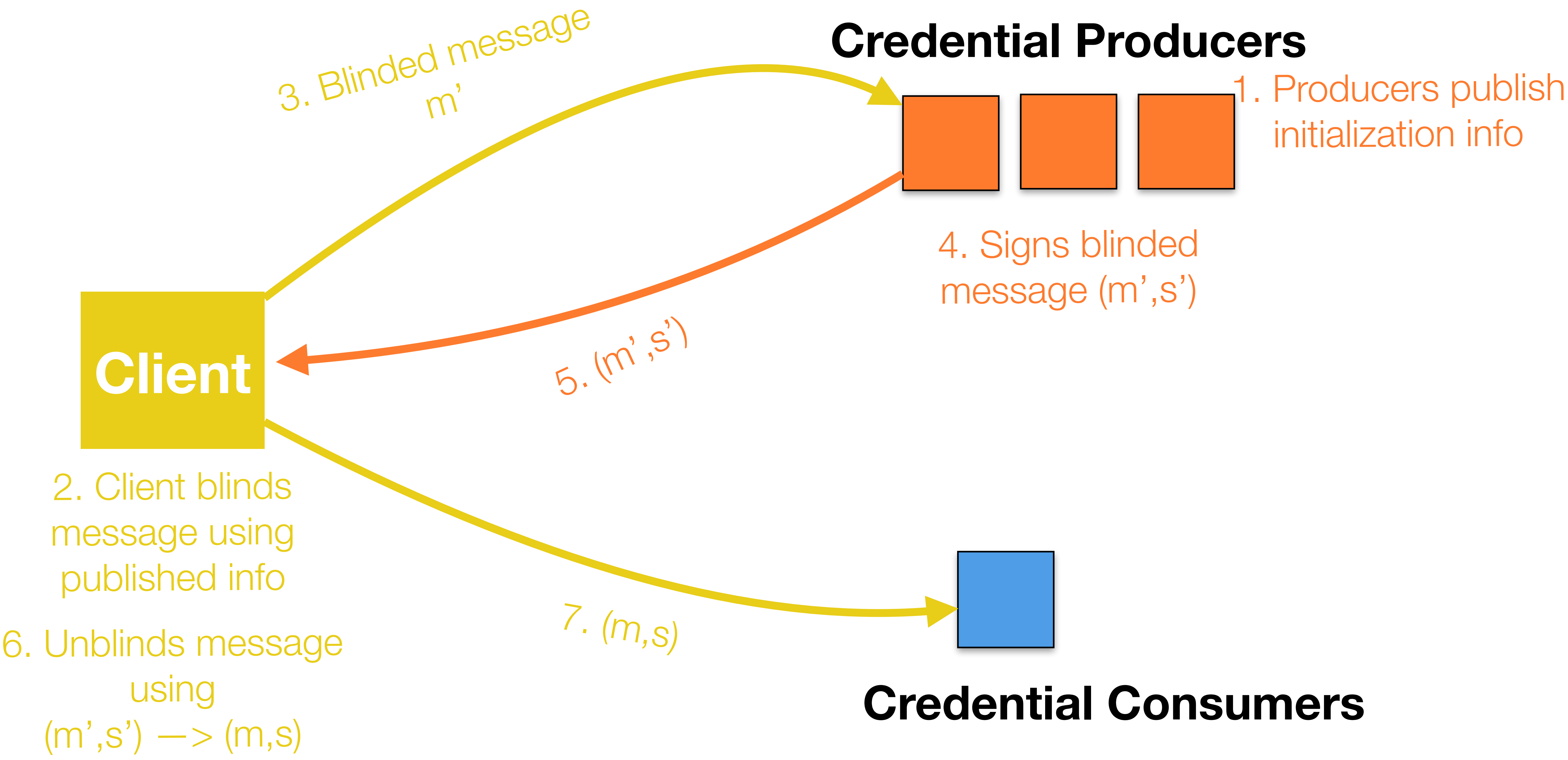
4. Signs blinded message (m’,s’)

5. (m’ ,s’)

6. Unblinds message using (m’,s’) —> (m,s)

Credential Consumers

At-Large Credential Scheme



Credential Producers

1. Producers publish initialization info

Client

2. Client blinds message using published info

3. Blinded message m’

4. Signs blinded message (m’,s’)

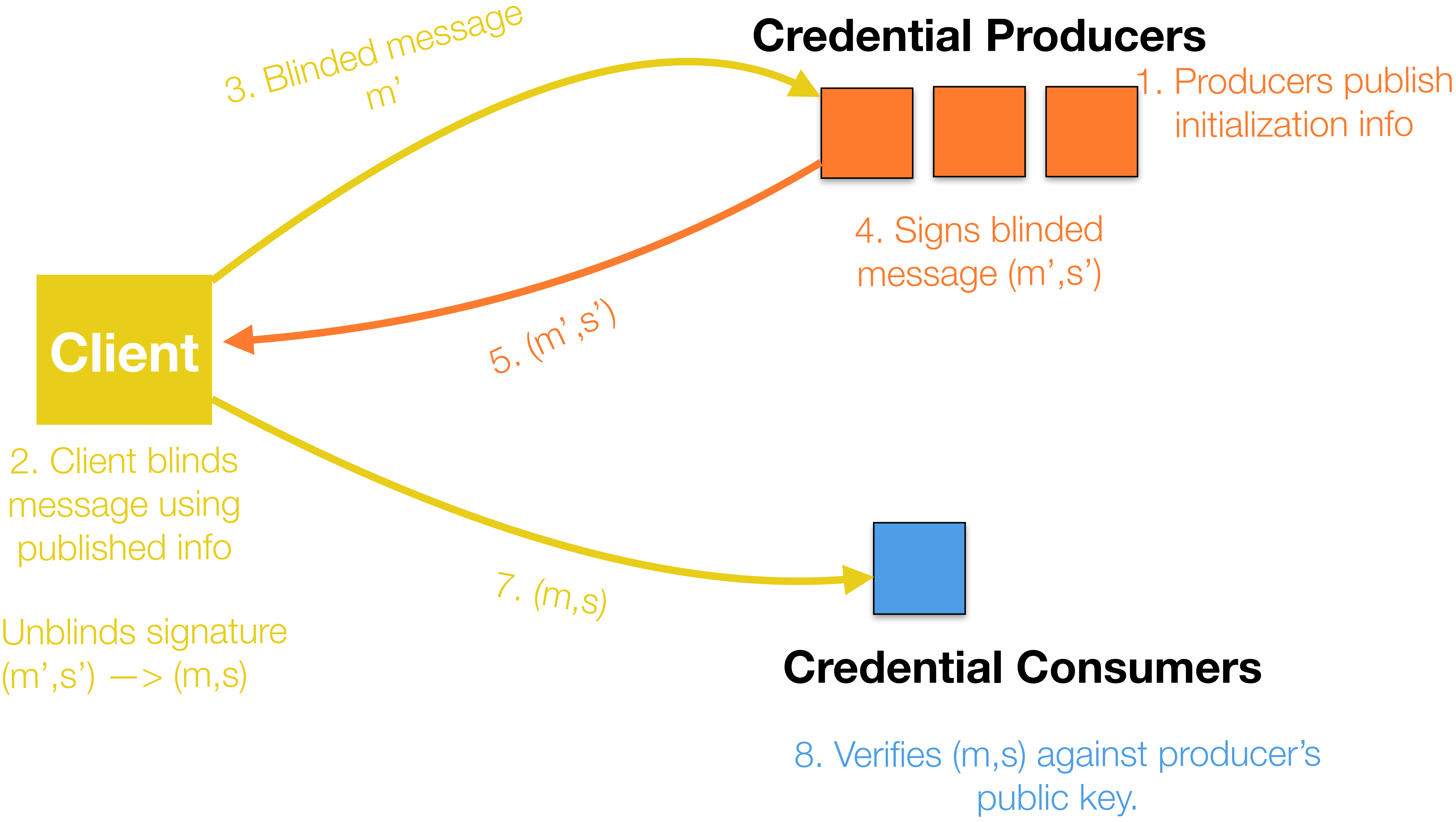
5. (m’,s’)

6. Unblinds message using (m’,s’) —> (m,s)

7. (m,s)

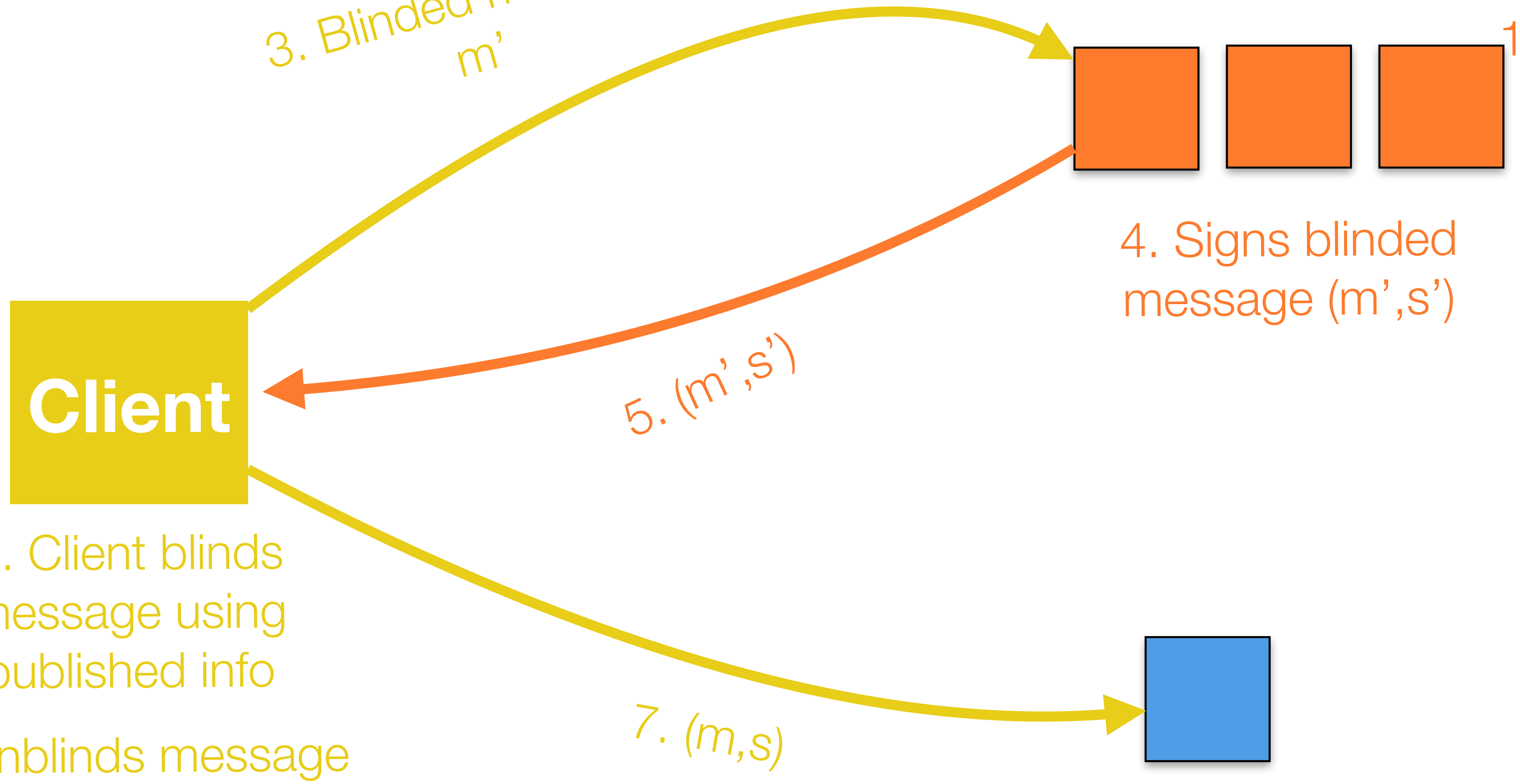
Credential Consumers

At-Large Credential Scheme



3. Blinded message m’
Credential Producers
1. Producers publish initialization info
4. Signs blinded message (m’,s’)
Client
5. (m’,s’)
2. Client blinds message using published info
6. Unblinds signature (m’,s’) —> (m,s)
7. (m,s)
Credential Consumers
8. Verifies (m,s) against producer’s public key.

At-Large Credential Scheme



Credential Producers

1. Producers publish initialization info

Client

2. Client blinds message using published info

3. Blinded message m’

4. Signs blinded message (m’,s’)

5. (m’,s’)

6. Unblinds message using (m’,s’) —> (m,s)

7. (m,s)

Credential Consumers

8. Verifies (m,s) against producer’s public key.

9. If (t,n) threshold is reached client is authenticated to application.

Roadmap

1. Background
2. Work Overview
3. System Architecture
4. Credential Producers and Consumers
   - At -Large Credentials
   - **Group Credentials**
5. Evaluation
6. Conclusions

Group Credential Scheme

Provides k-anonymous authentication

Verifiable whistleblowing/private chat room use cases

Group Credential Scheme

- Allows a client to authenticate explicitly as some member of a larger, well defined set of users (e.g. a Facebook group)

- The group credential scheme provides k-anonymity, the client is anonymous among a set of k people

- Based on linkable ring signatures

Technical Building Block: Linkable Ring Signatures

- Created by member of a group of users

- Third party can verify:
  - ***Some*** member of the group created signature
  - Whether two signatures were created by same signer

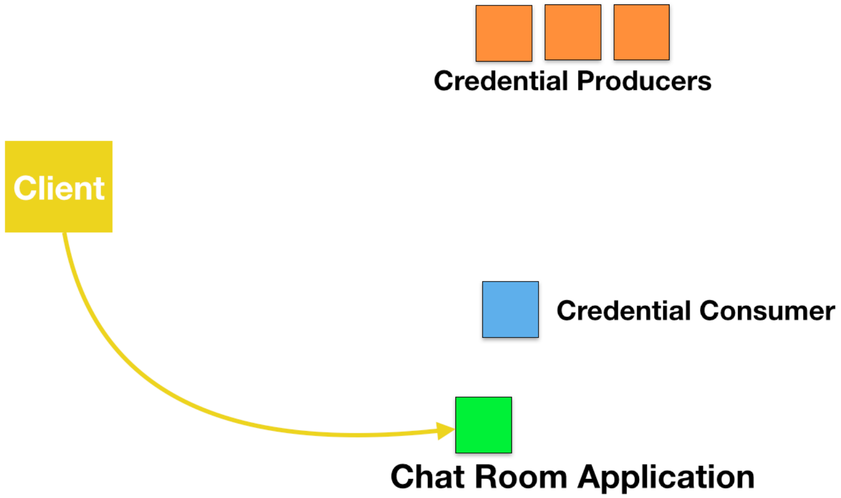
- Third party cannot discover
  - Which specific user created the signature

Technical Building Block: Linkable Ring Signatures

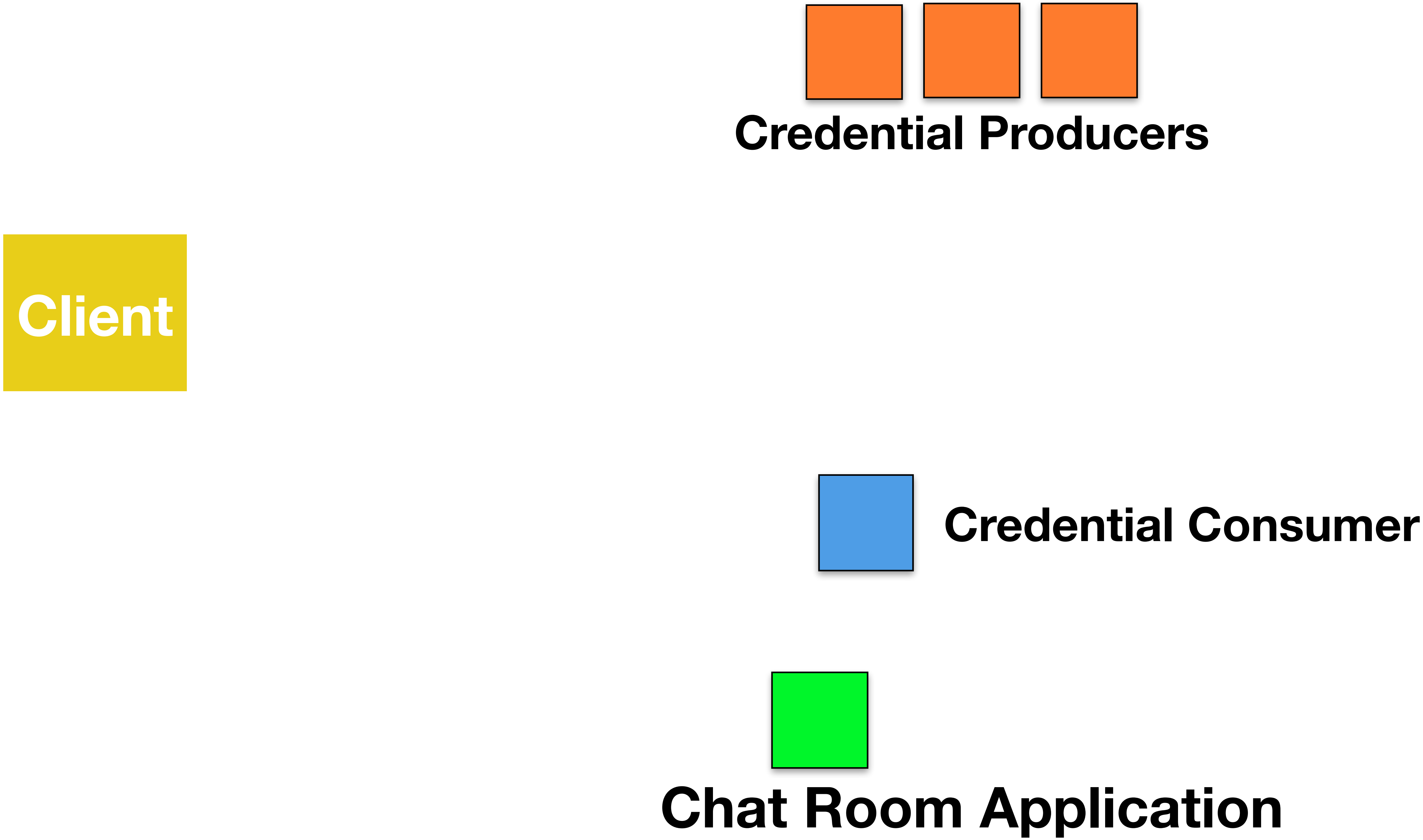
- LRS has *linkage tag*
  - If a client generates two LRSs, will have the same linkage tag
  - Means LRSs can be linked across time

- Linkage tag provides *accountability*
  - privacy preserving mapping between fed IDs and pseudonyms

Group Setup


Credential Producers
Client
Credential Consumer
Chat Room Application

Group Credential Scheme



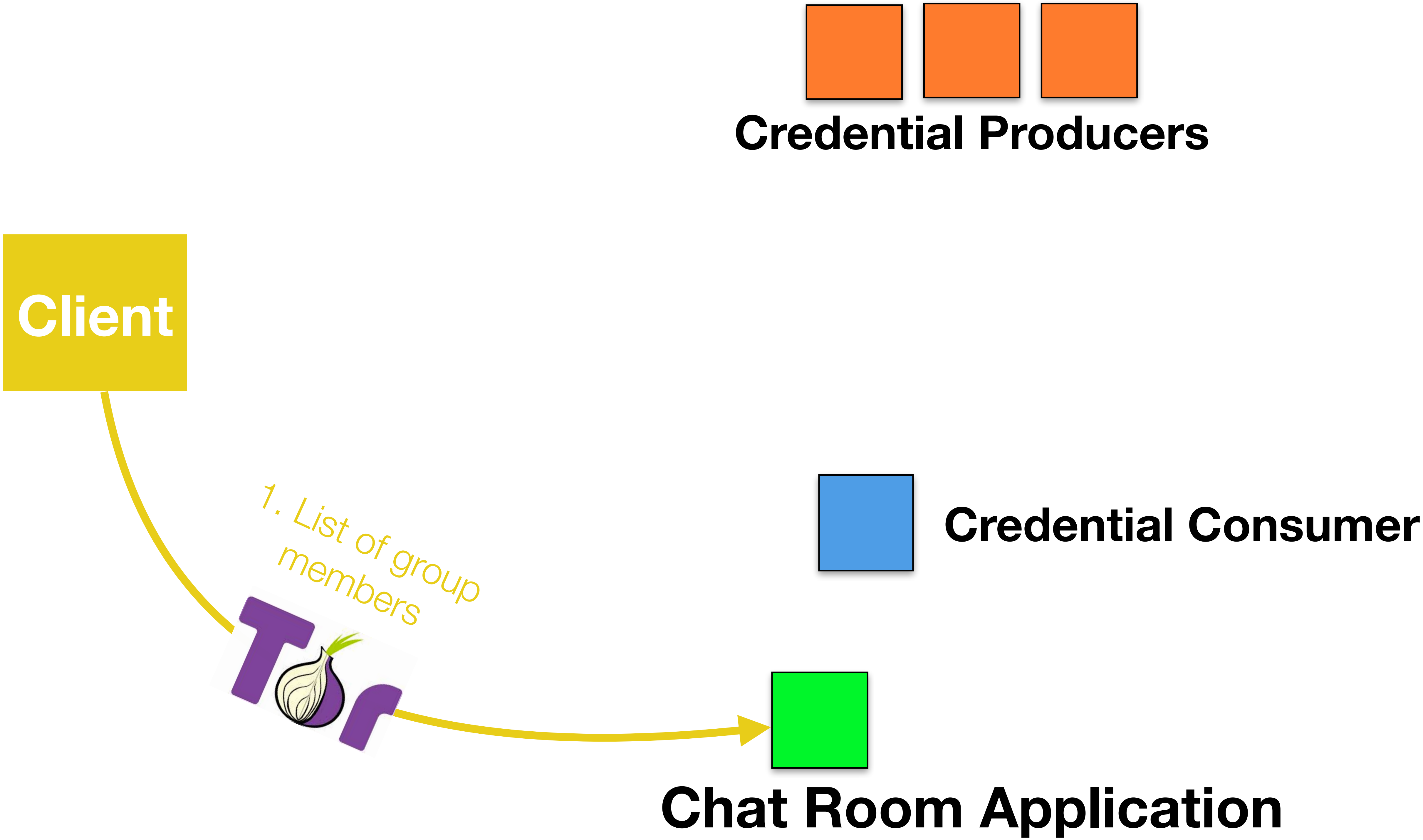
Credential Producers

Client

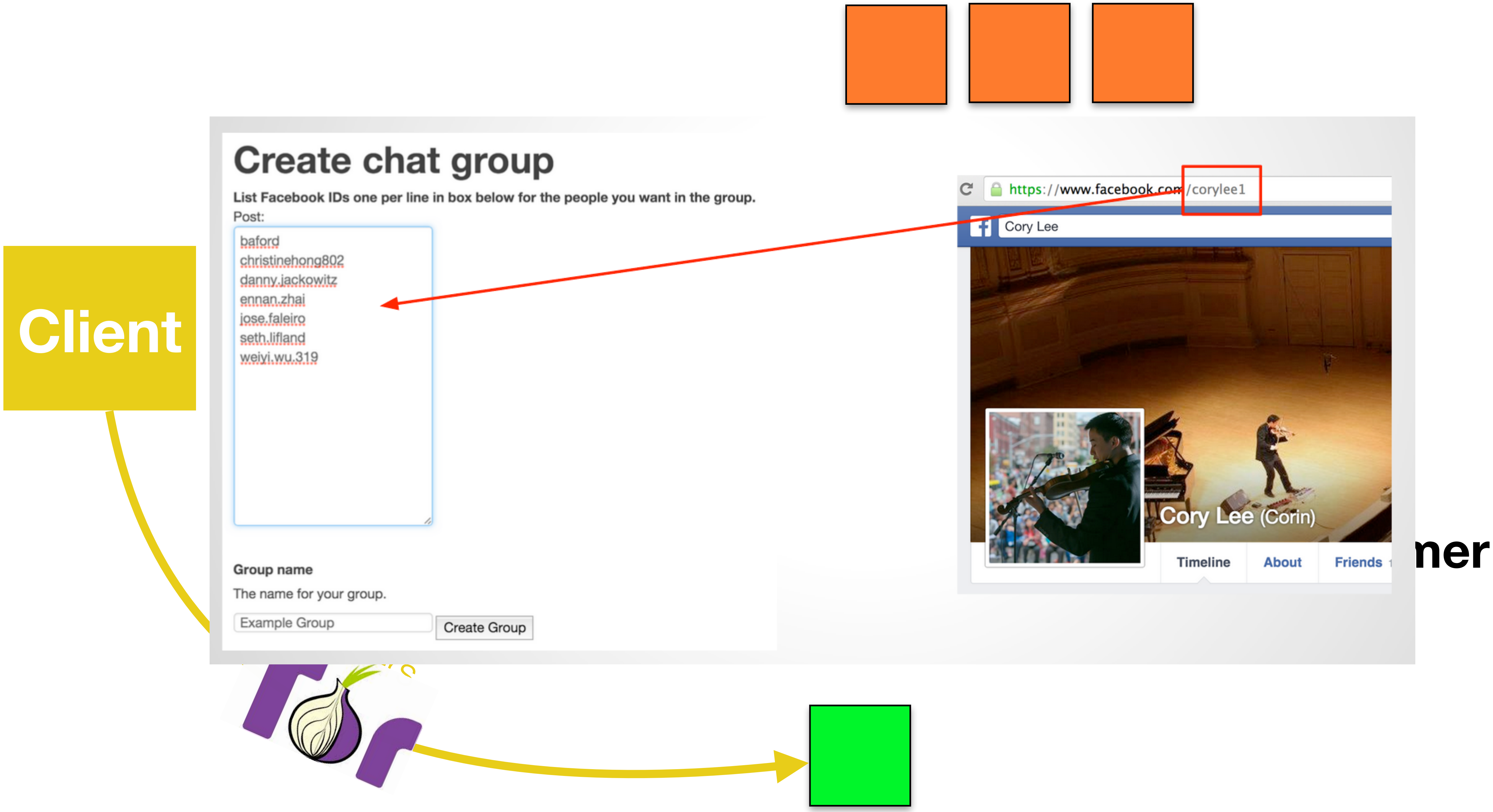
Credential Consumer

Chat Room Application

Group Credential Scheme



Group Credential Scheme



Create chat group

List Facebook IDs one per line in box below for the people you want in the group.

Post:

baford
christinehong802
danny.jackowitz
ennan.zhai
jose.faleiro
seth.lifland
weiyi.wu.319

Group name

The name for your group.

Example Group

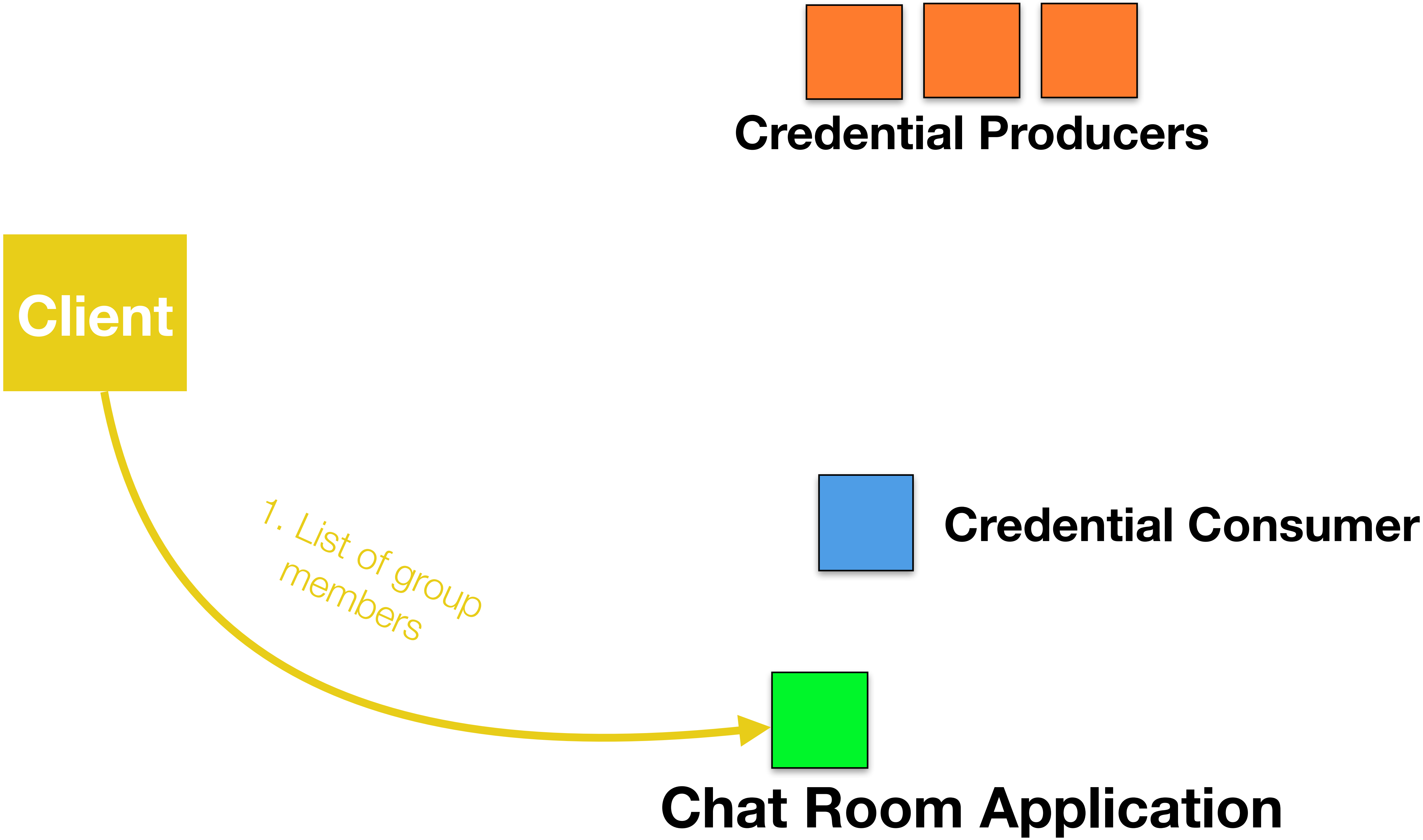
https://www.facebook.com/corylee1

Cory Lee

Cory Lee (Corin)

Chat Room Application

Group Credential Scheme



Credential Producers

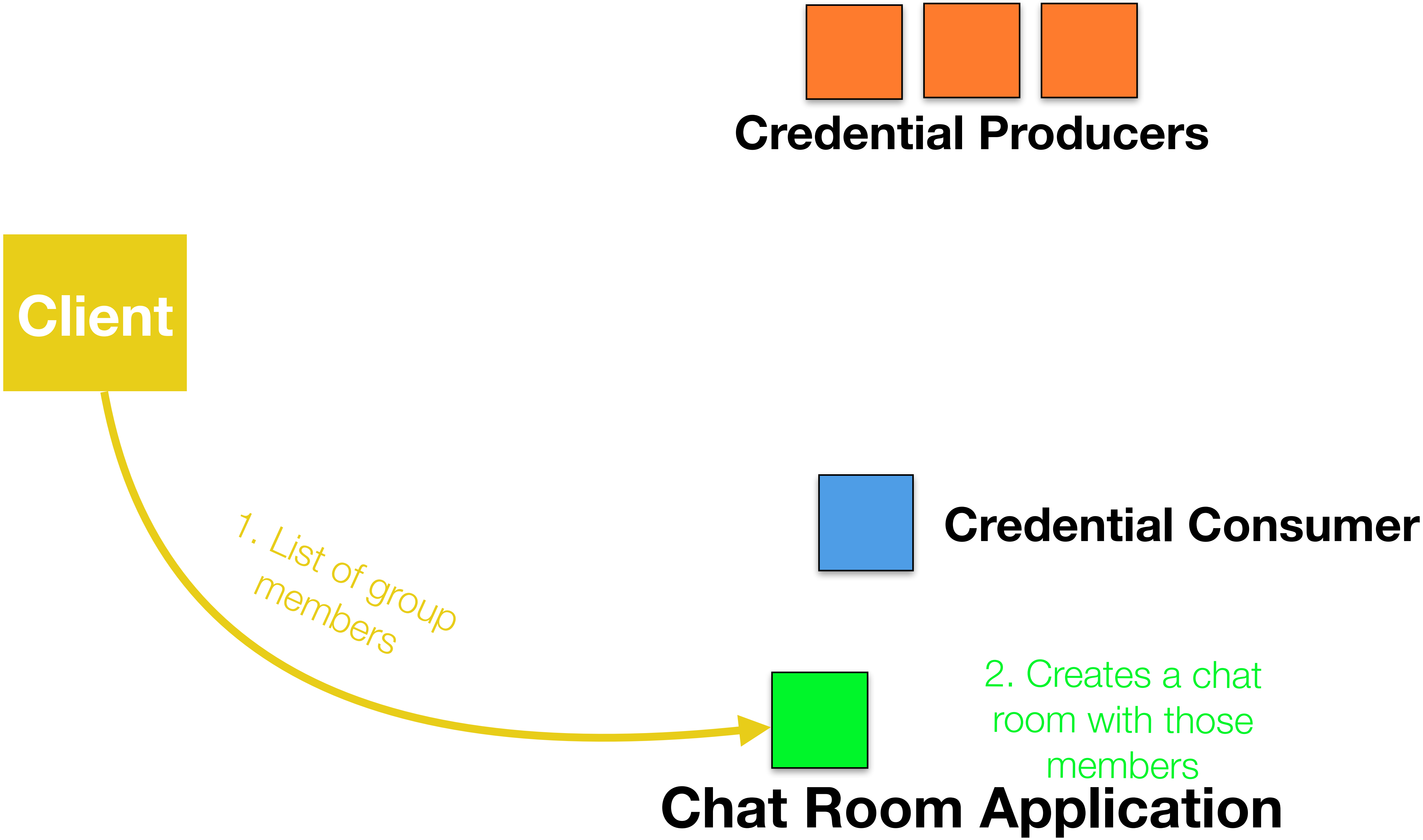
Client

1. List of group members

Credential Consumer

Chat Room Application

Group Credential Scheme



Credential Producers

Client

1. List of group members

Credential Consumer

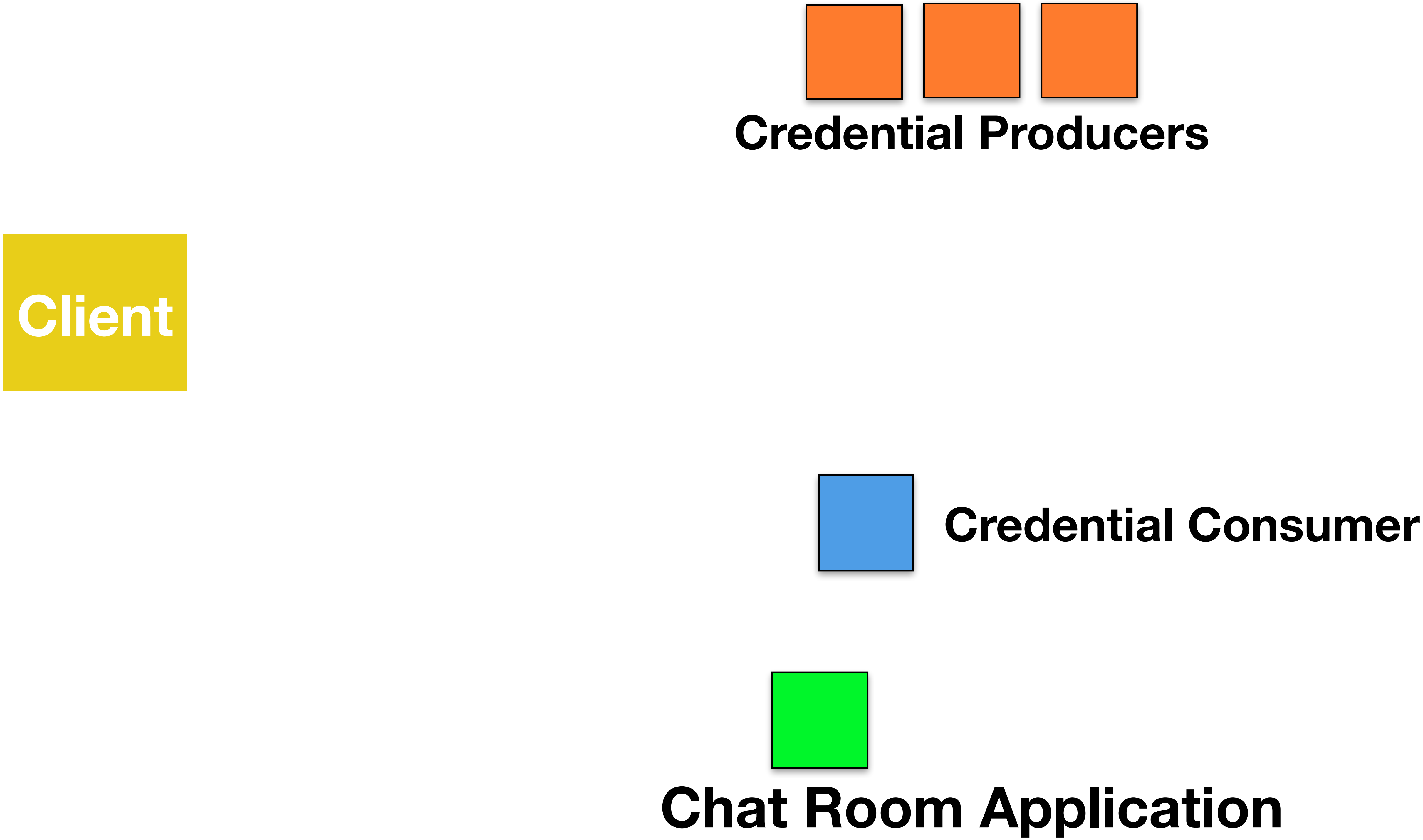
2. Creates a chat room with those members

Chat Room Application

Group Credential Scheme

- The client collects their private key shares from at least t of n credential producers
- Client combines shares to give private key, saved in browser extension
- Client collects public keys from credential producers (no authentication)
- Credential consumers issue challenge to client, which client signs with LRS and is the authenticated to application

Group Credential Scheme



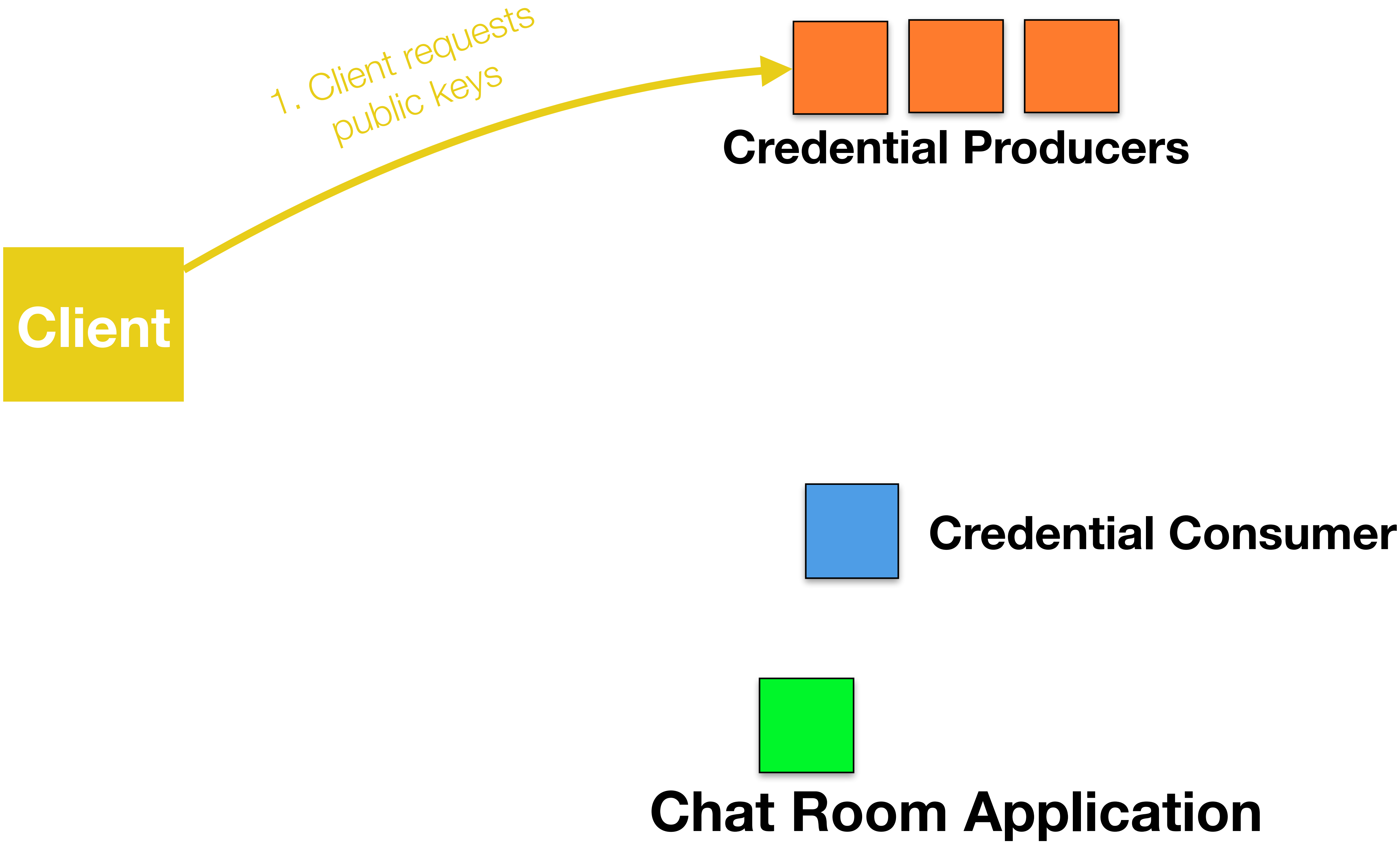
Credential Producers

Client

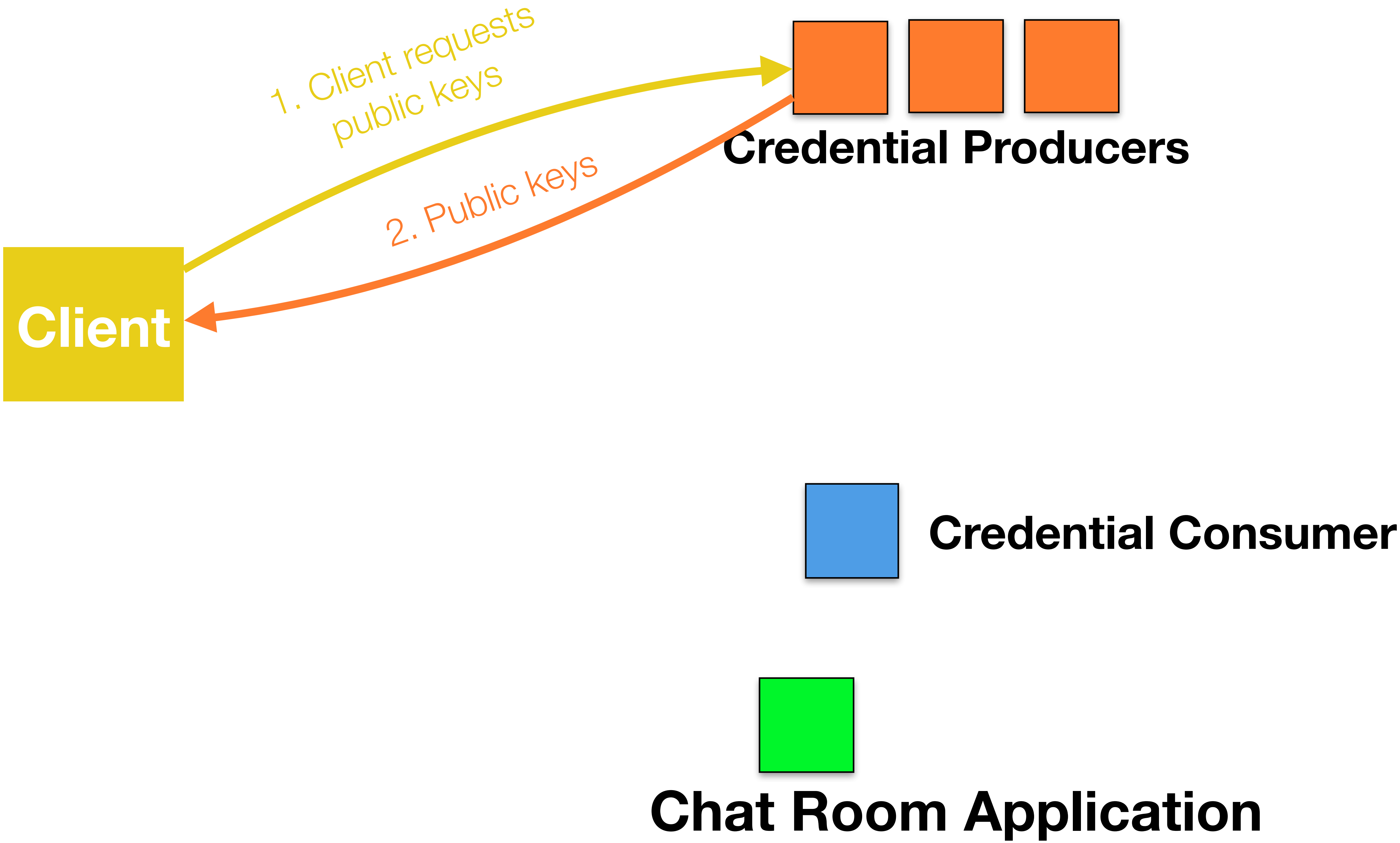
Credential Consumer

Chat Room Application

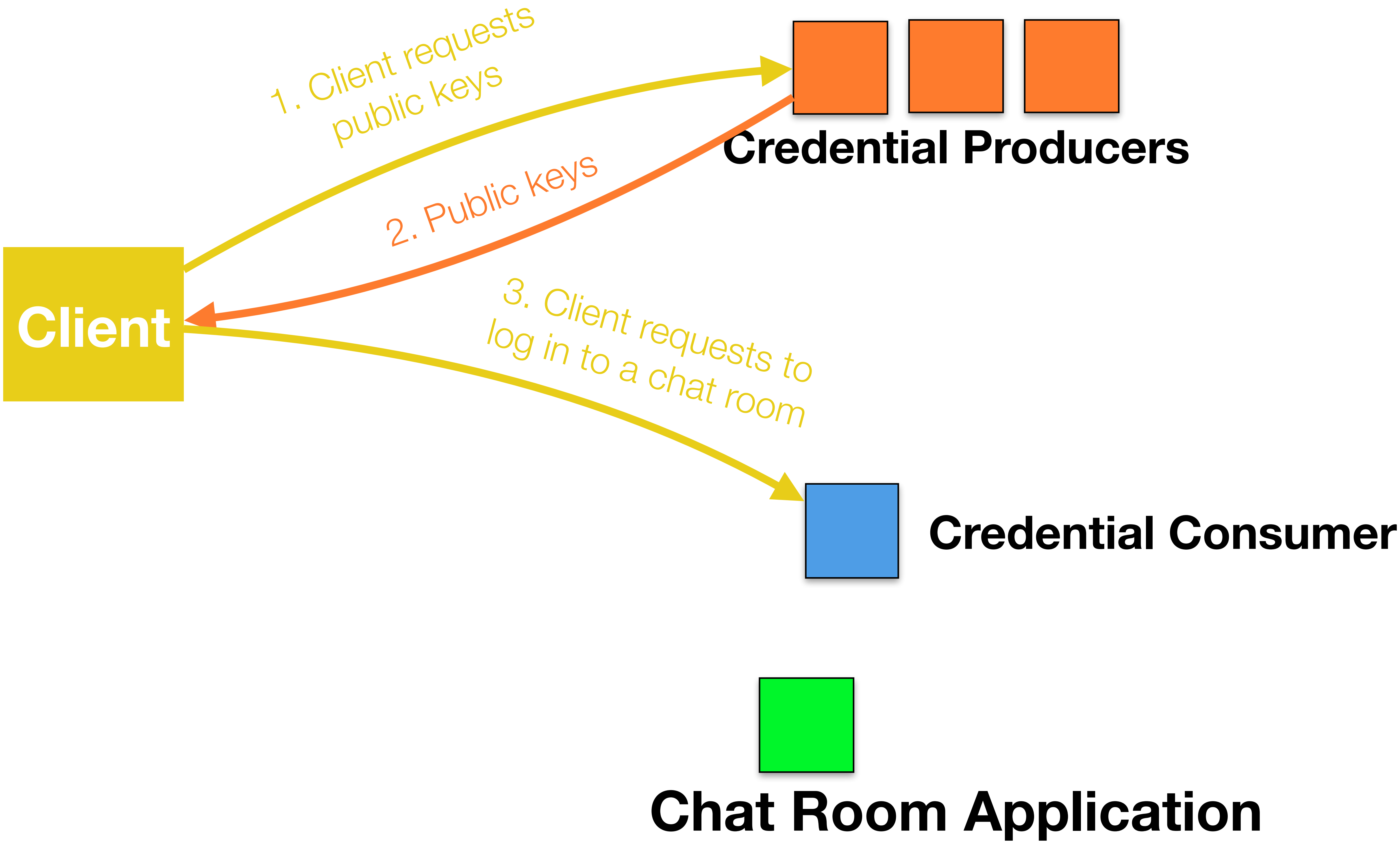
Group Credential Scheme



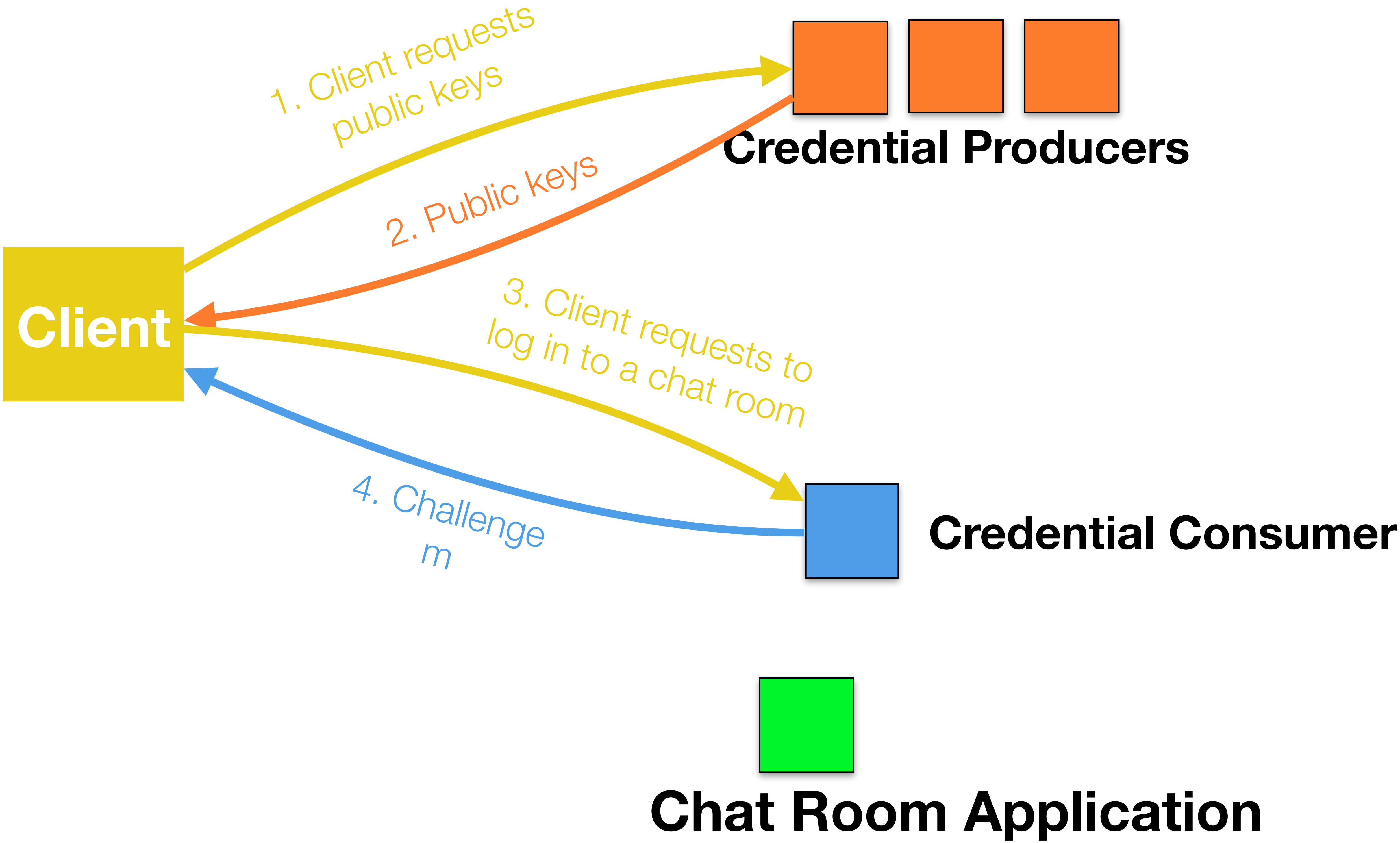
Group Credential Scheme



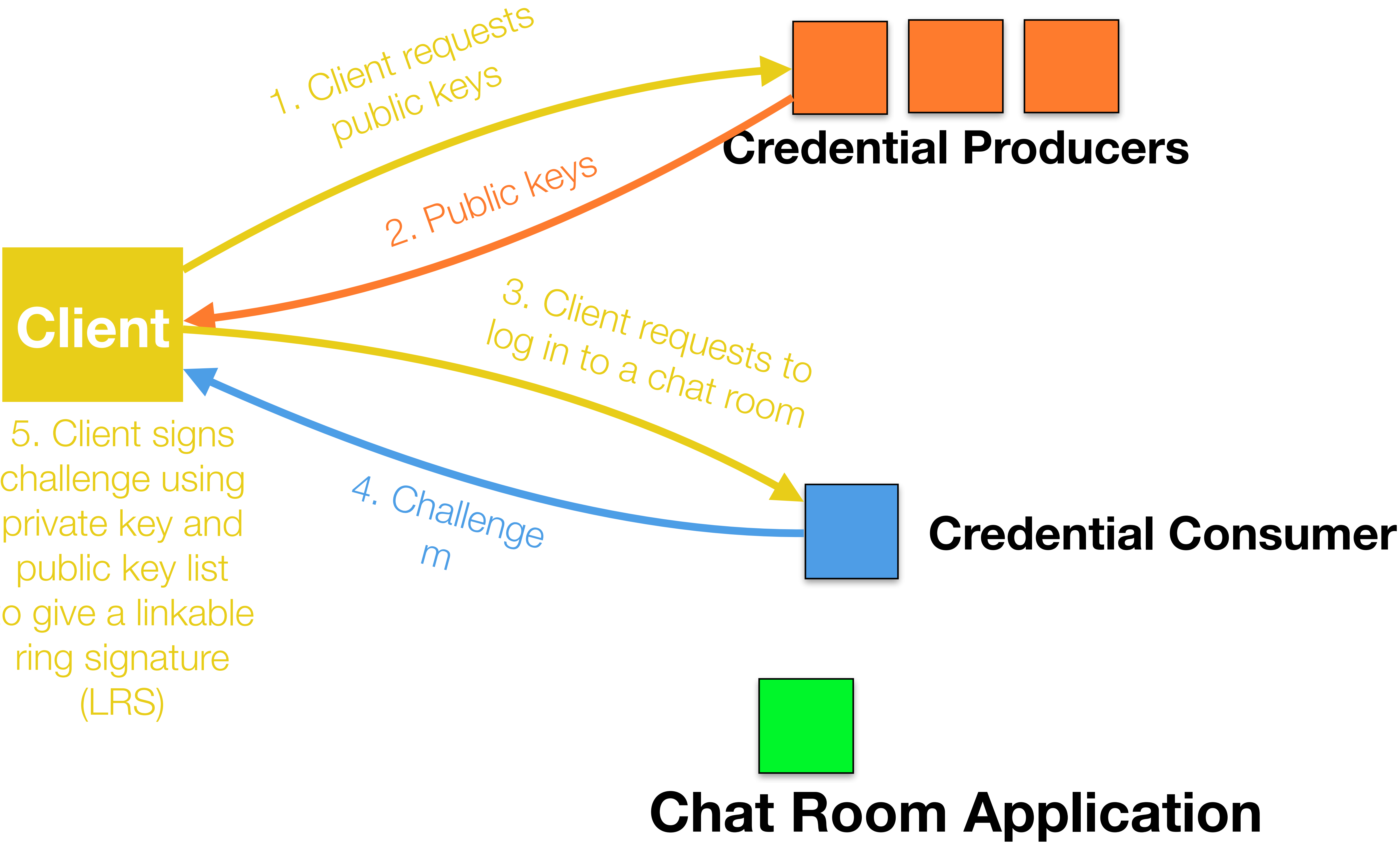
Group Credential Scheme



Group Credential Scheme



Group Credential Scheme



1. Client requests public keys

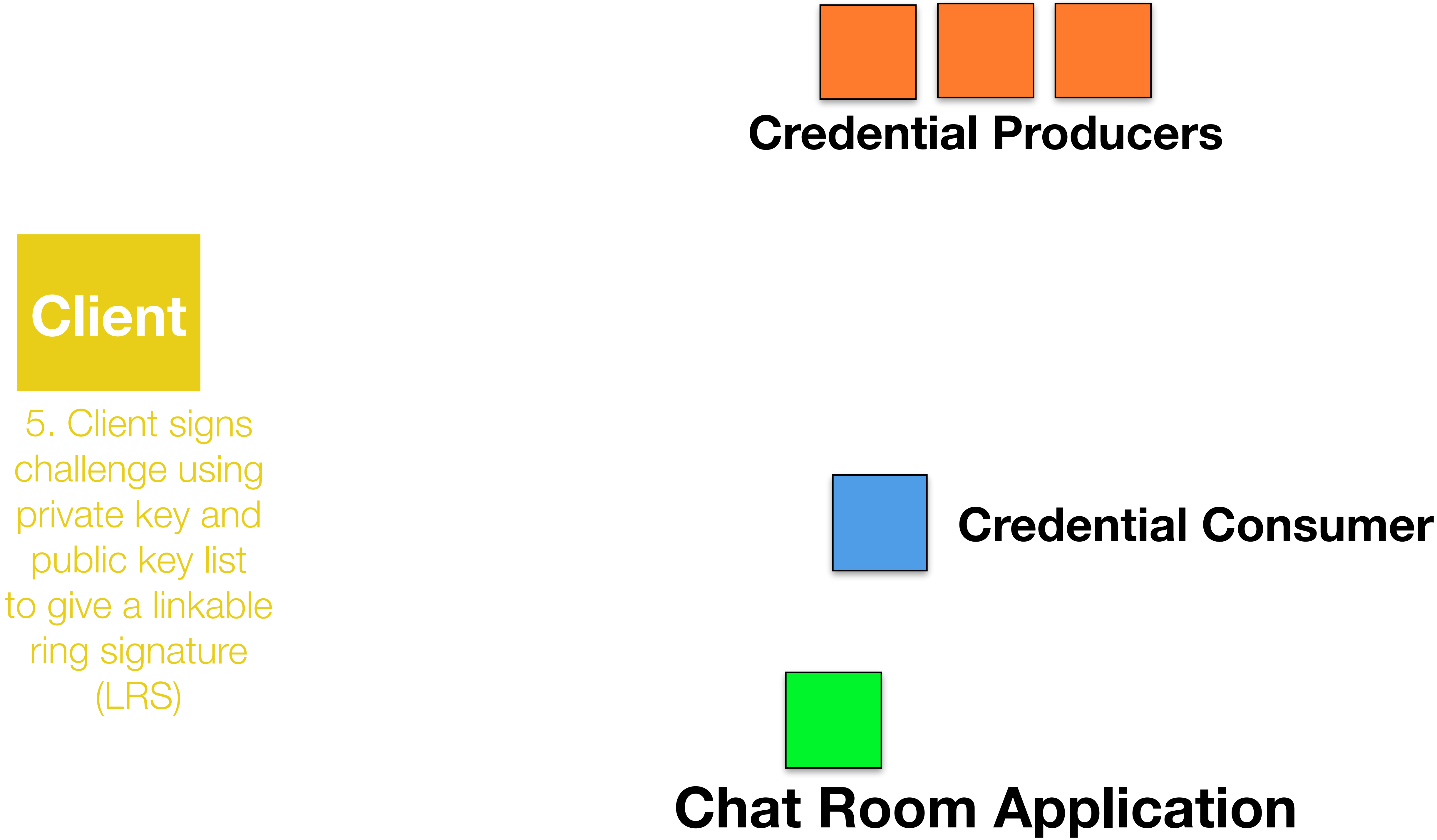
2. Public keys

3. Client requests to log in to a chat room

4. Challenge m

5. Client signs challenge using private key and public key list to give a linkable ring signature (LRS)

Group Credential Scheme



Credential Producers

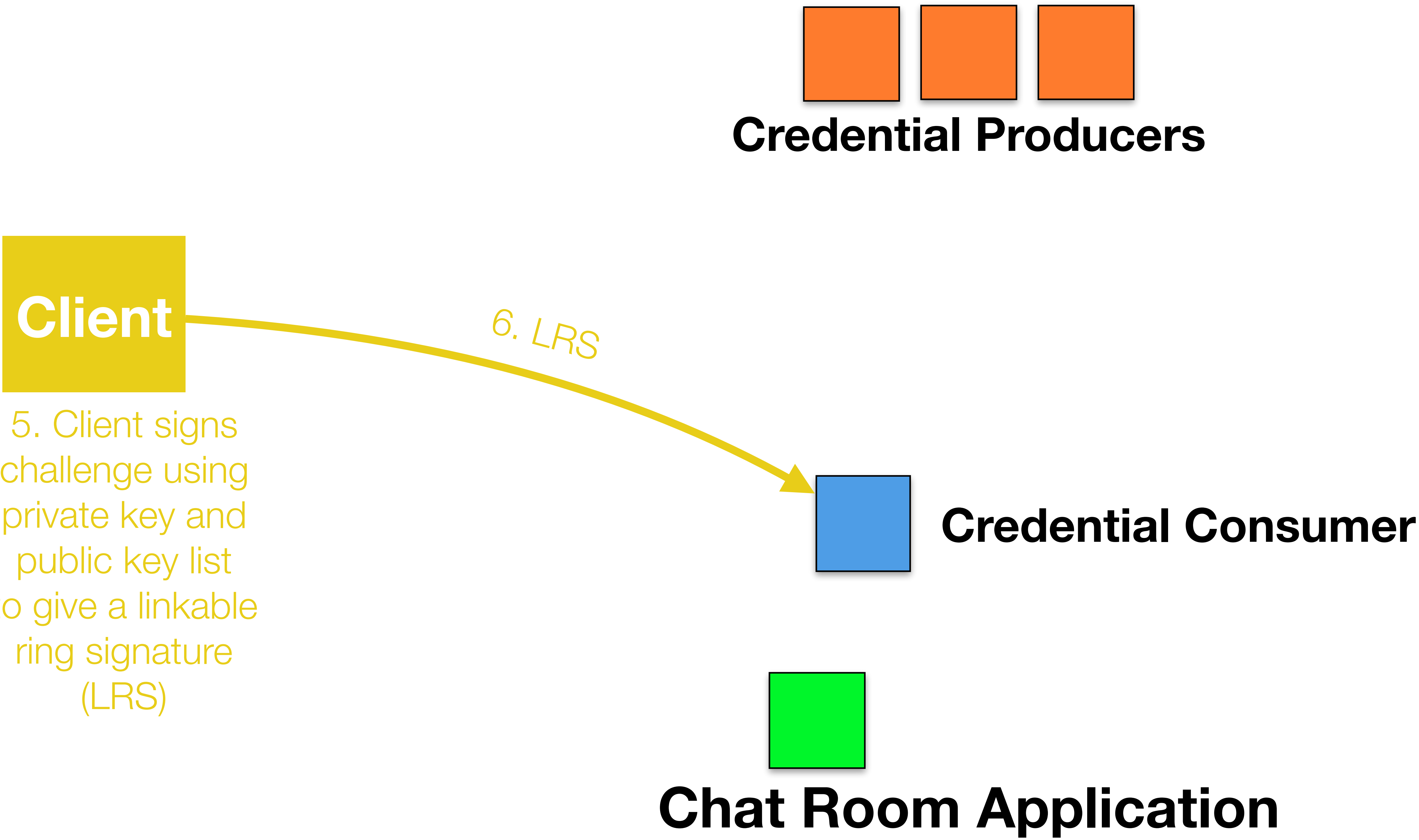
Client

5. Client signs challenge using private key and public key list to give a linkable ring signature (LRS)

Credential Consumer

Chat Room Application

Group Credential Scheme



Credential Producers

Client

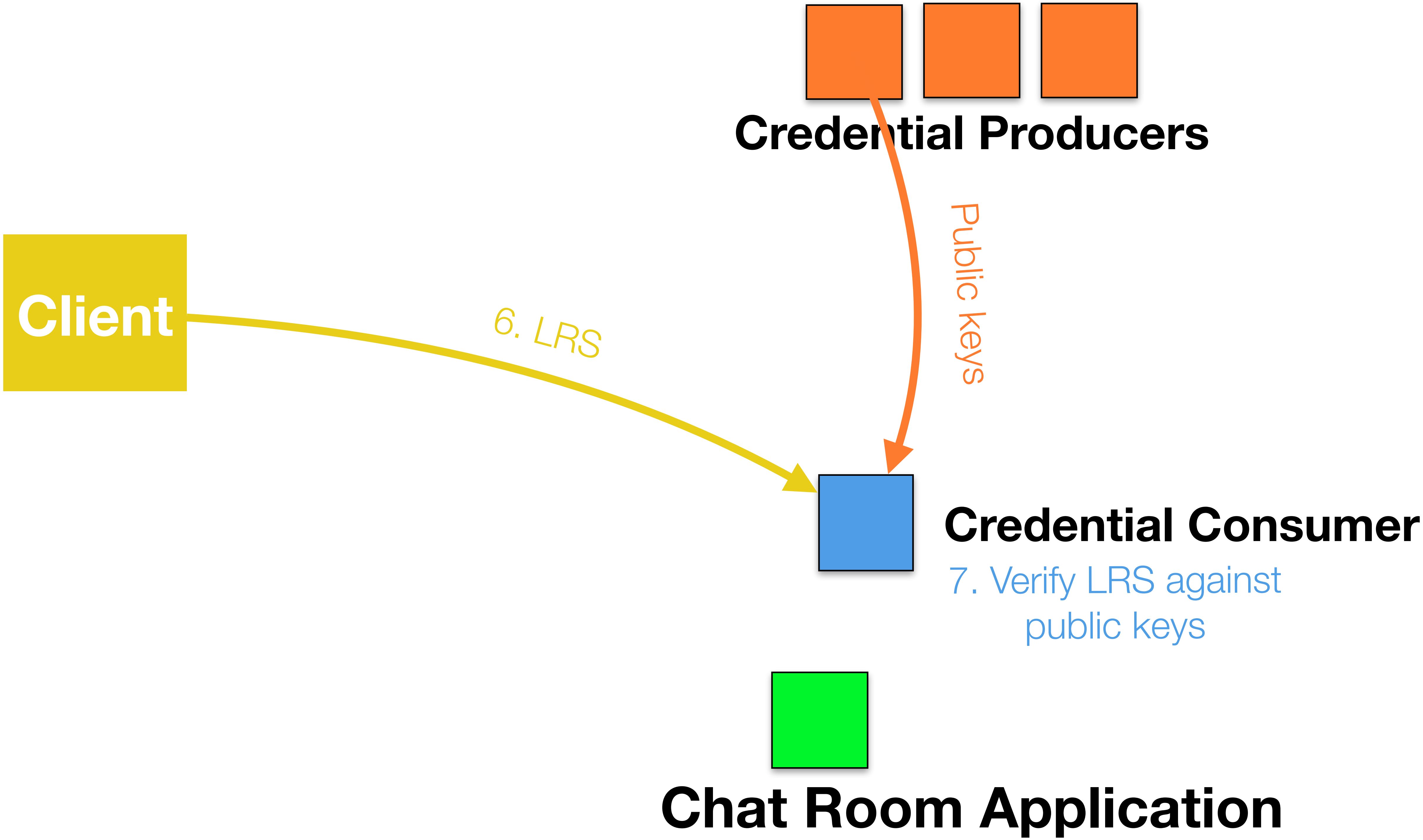
6. LRS

5. Client signs challenge using private key and public key list to give a linkable ring signature (LRS)

Credential Consumer

Chat Room Application

Group Credential Scheme



Credential Producers

Public keys

Client

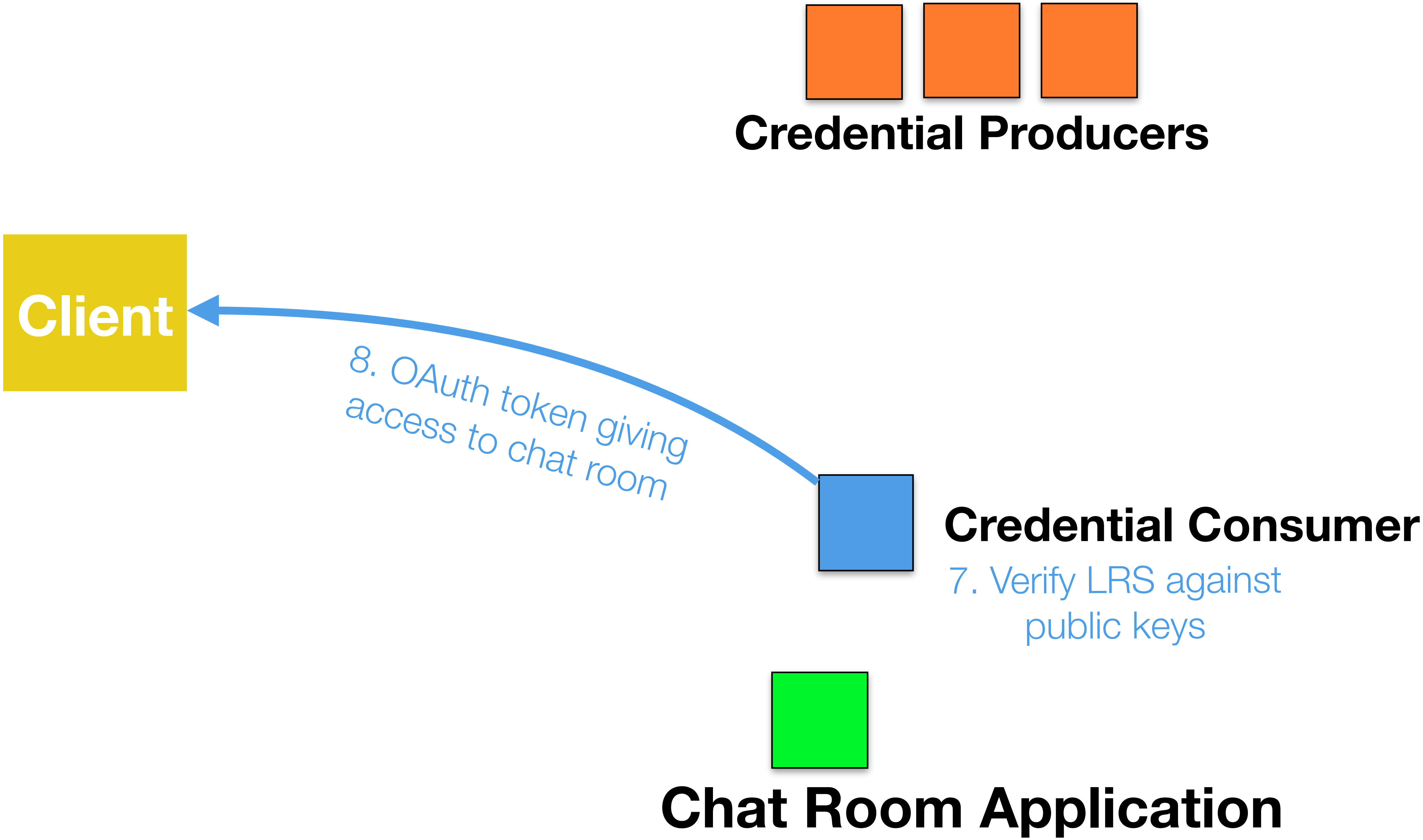
6. LRS

Credential Consumer

7. Verify LRS against public keys

Chat Room Application

Group Credential Scheme



Credential Producers

Client

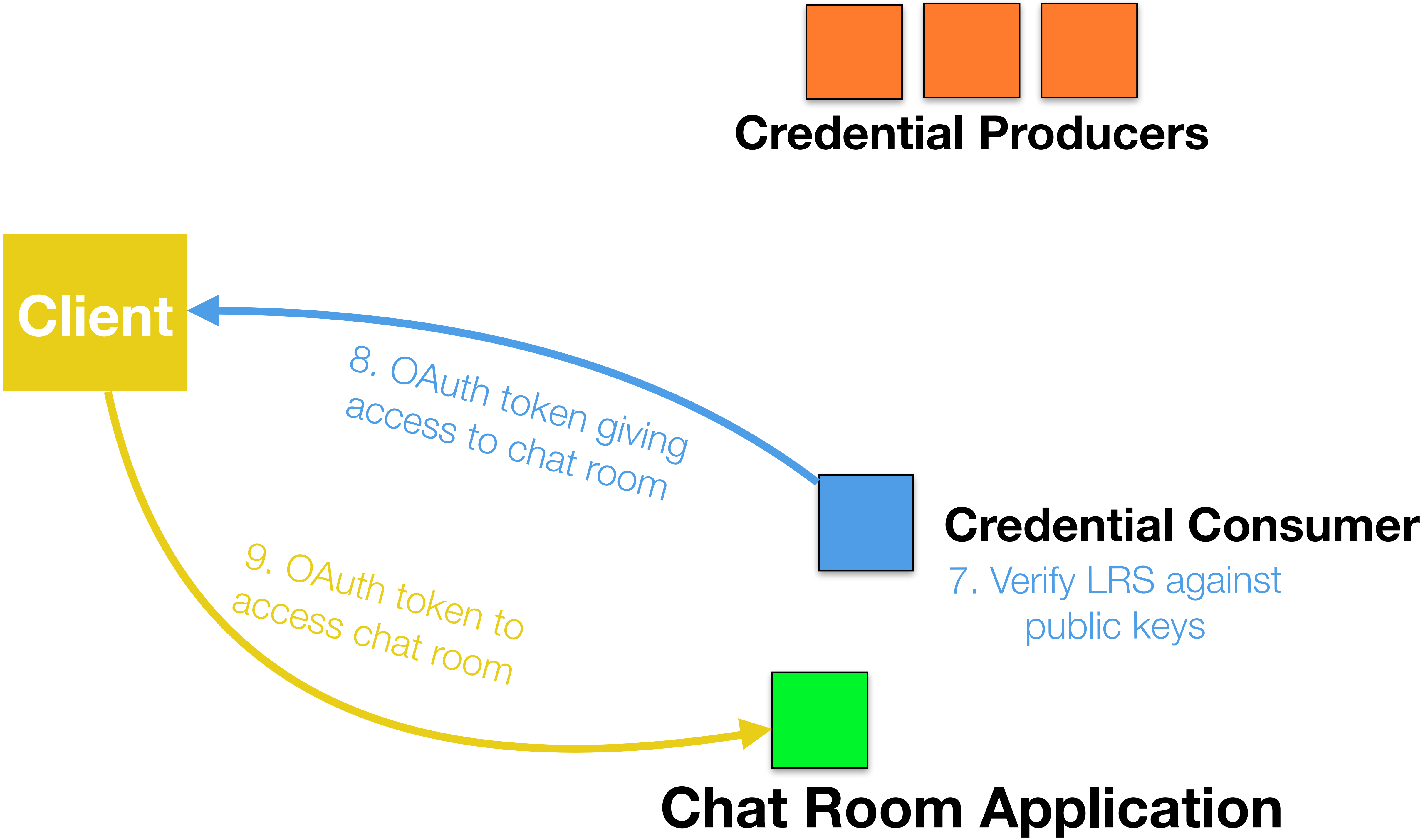
8. OAuth token giving access to chat room

Credential Consumer

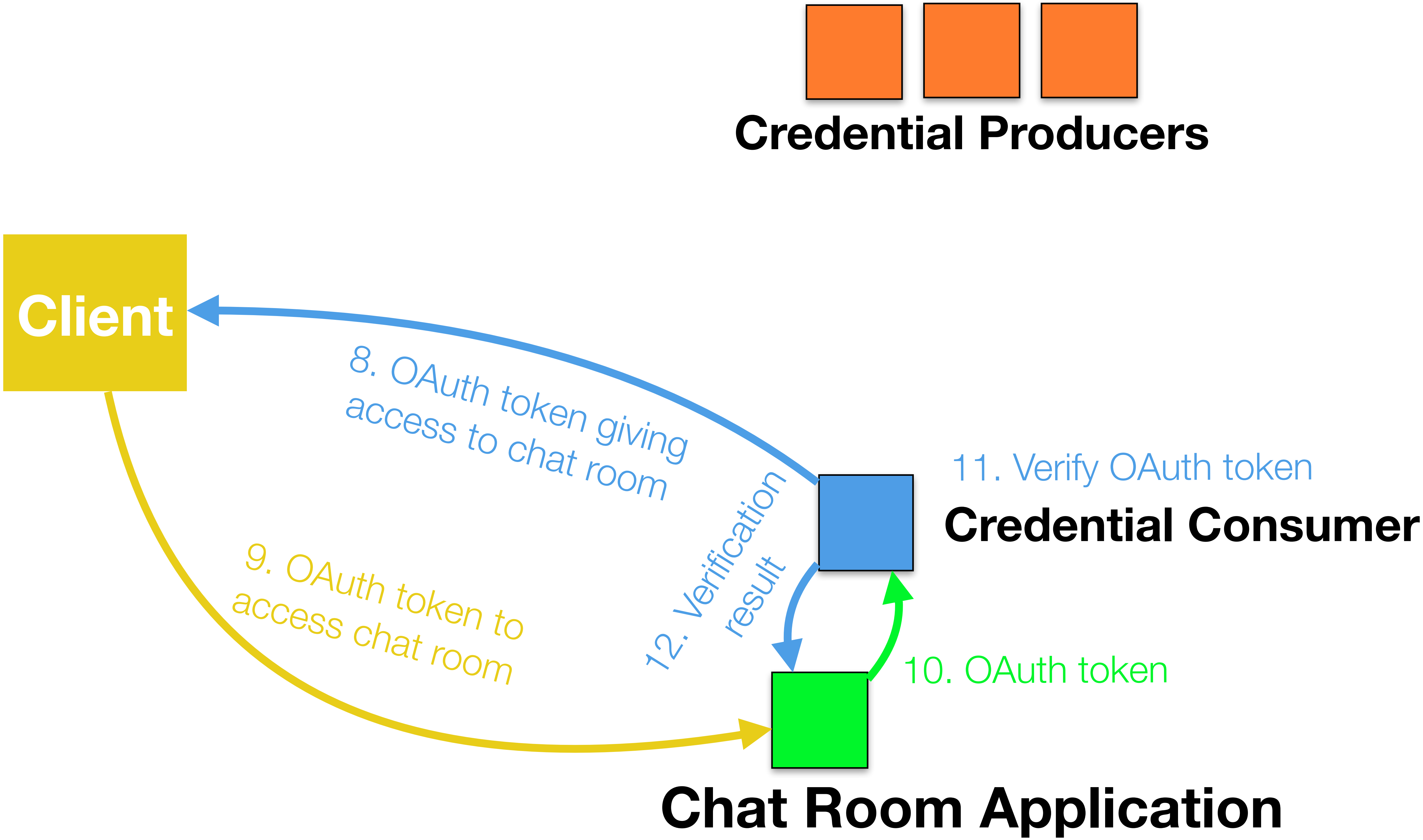
7. Verify LRS against public keys

Chat Room Application

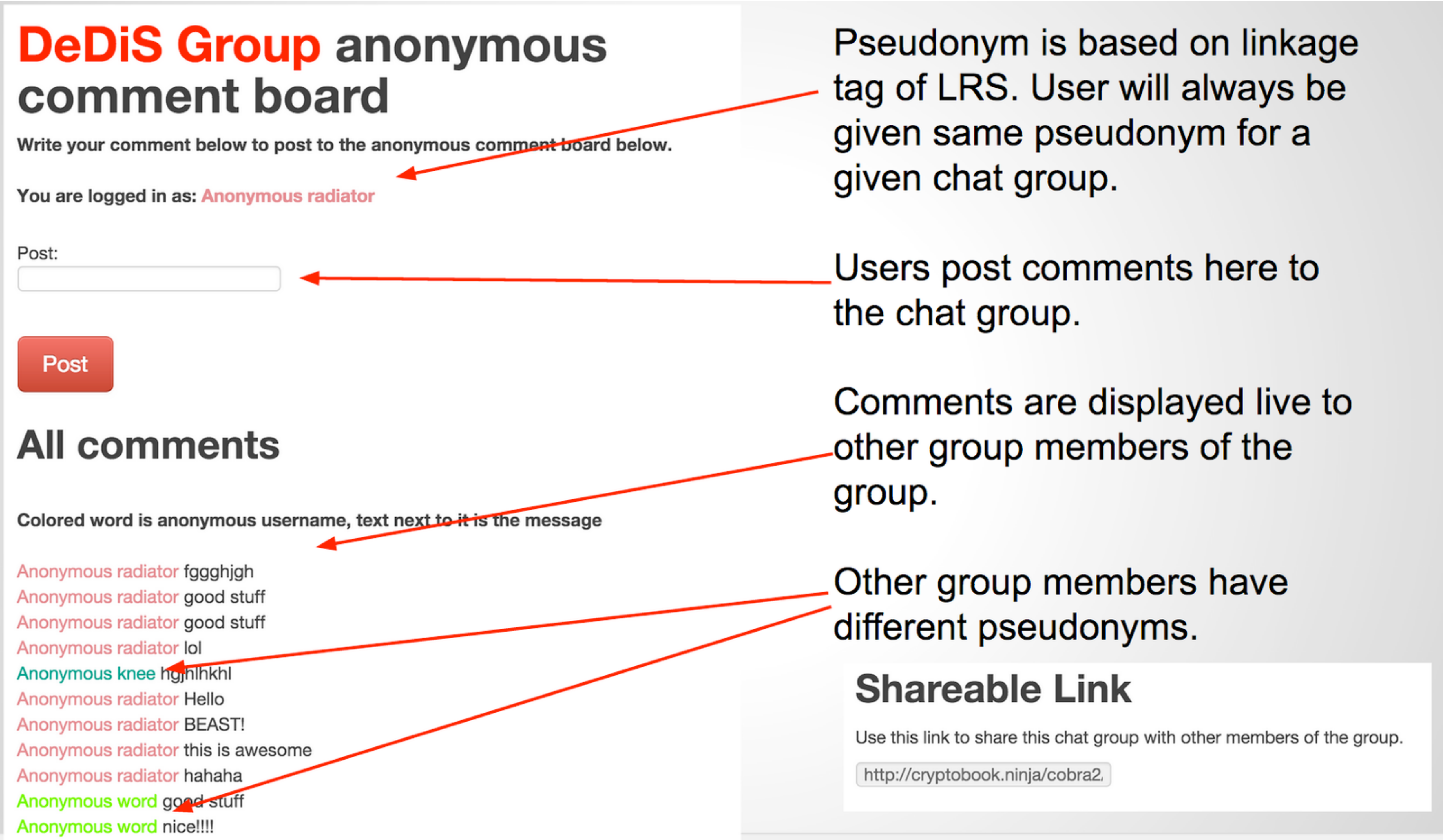
Group Credential Scheme


Credential Producers
Client
8. OAuth token giving access to chat room
Credential Consumer
7. Verify LRS against public keys
9. OAuth token to access chat room
Chat Room Application

Group Credential Scheme



Group Credential Scheme: Chat Room



DeDiS Group anonymous comment board

Write your comment below to post to the anonymous comment board below.

You are logged in as: Anonymous radiator

Post:

All comments

Colored word is anonymous username, text next to it is the message

Anonymous radiator fggghjgh
Anonymous radiator good stuff
Anonymous radiator good stuff
Anonymous radiator lol
Anonymous knee hgjhlhkhl
Anonymous radiator Hello
Anonymous radiator BEAST!
Anonymous radiator this is awesome
Anonymous radiator hahaha
Anonymous word good stuff
Anonymous word nice!!!!

Pseudonym is based on linkage tag of LRS. User will always be given same pseudonym for a given chat group.

Users post comments here to the chat group.

Comments are displayed live to other group members of the group.

Other group members have different pseudonyms.

Shareable Link

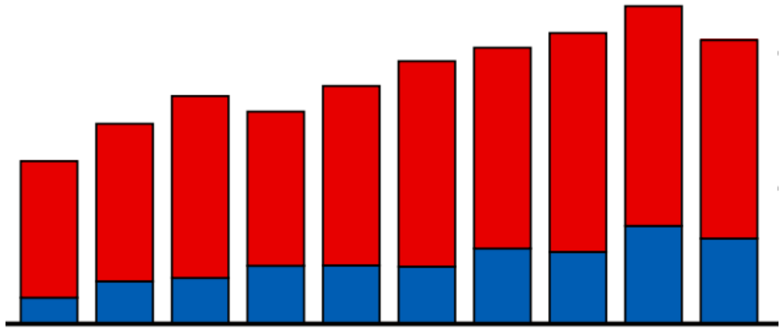
Use this link to share this chat group with other members of the group.

http://cryptobook.ninja/cobra2.

Roadmap

1. Background
2. Work Overview
3. System Architecture
4. Credential Producers and Consumers
    - At -Large Credentials
    - Group Credentials
5. **Evaluation**
6. Conclusions

Evaluation

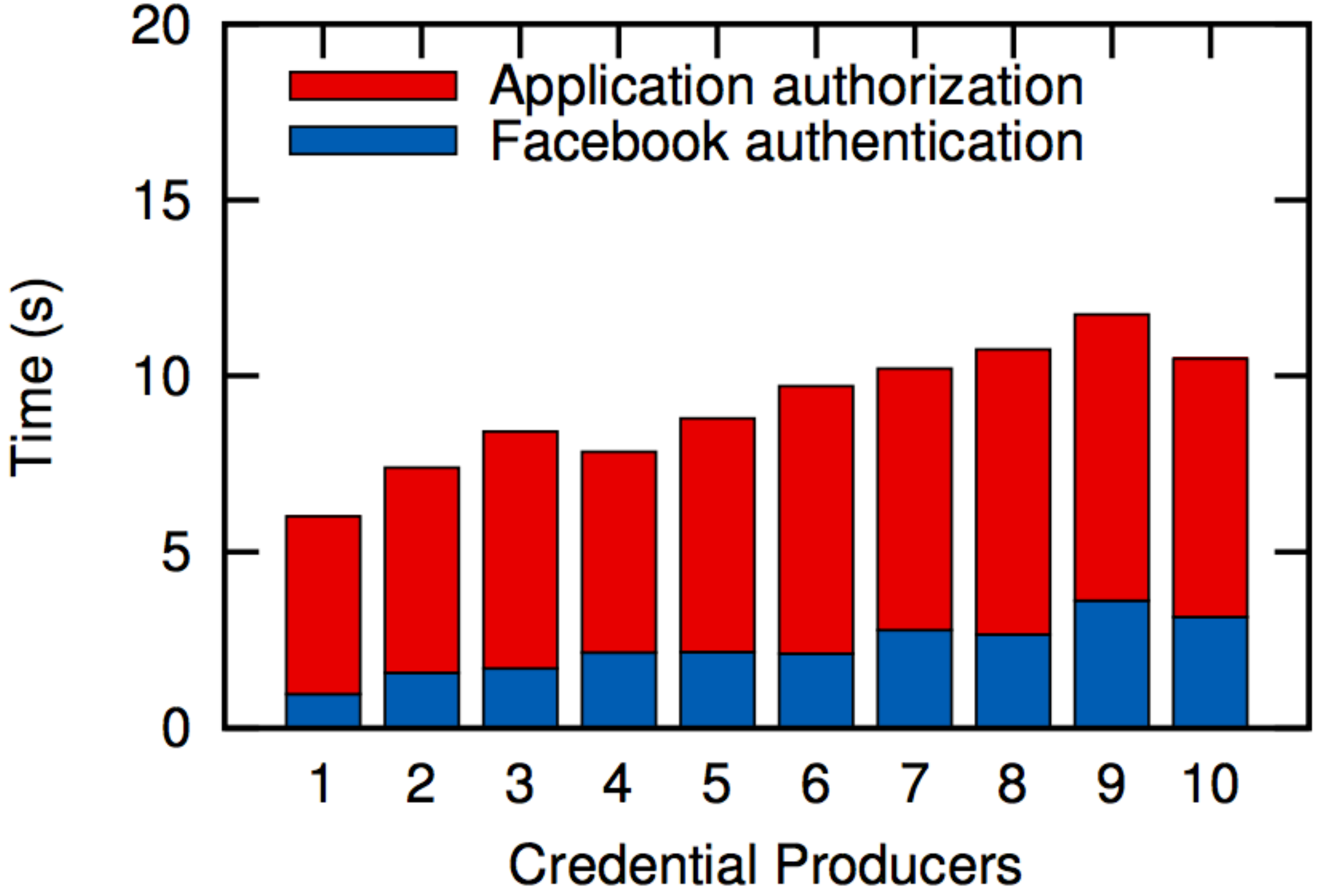


Evaluation: Experimental Setup

- **Clients:** consumer laptops
  - 2.4GHz Intel Core i5 processors
  - 8GB of RAM.
- **Credential producers:** PlanetLab nodes
  - 2.4GHz Intel Xeon processor
  - 4GB of RAM
- **Credential consumers:** commercial shared hosting
  - 2.4GHz Intel Xeon processors
  - 16GB of RAM

Evaluation: Producing Credentials, App Auth.

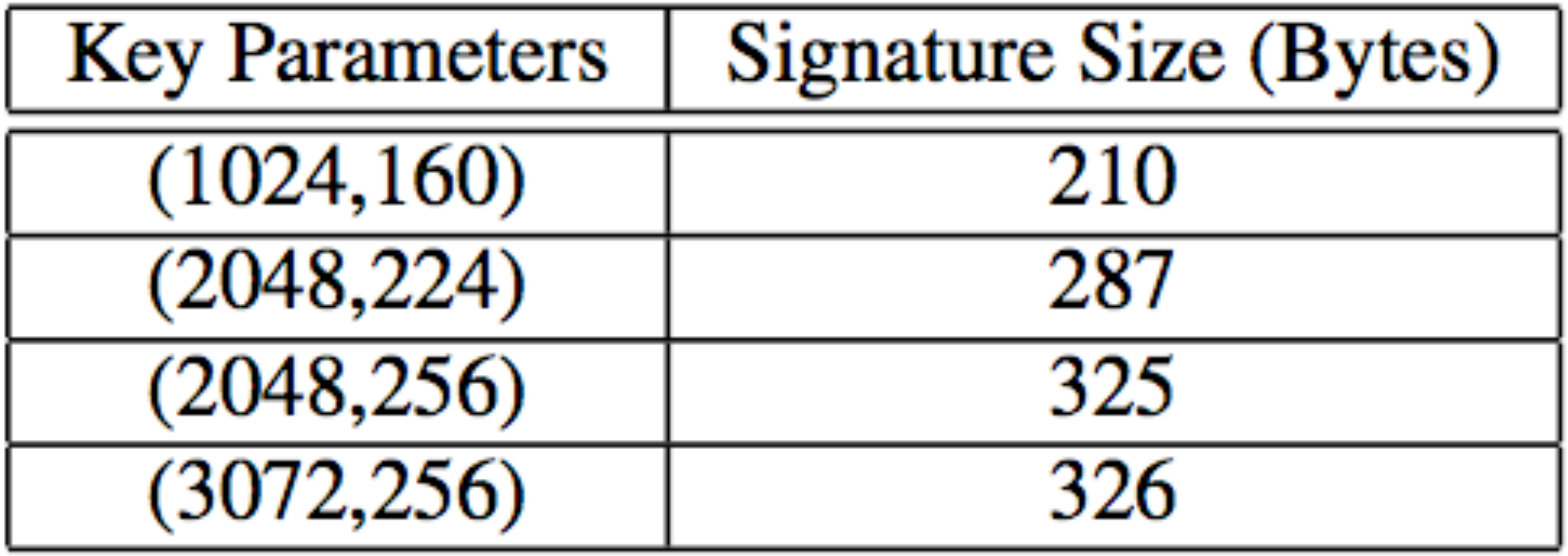
Facebook App Authorization time



- Client performs this setup step only once, the first time they use the system

Evaluation: Producing At-large Credentials

Blind Signature Size (bandwidth)

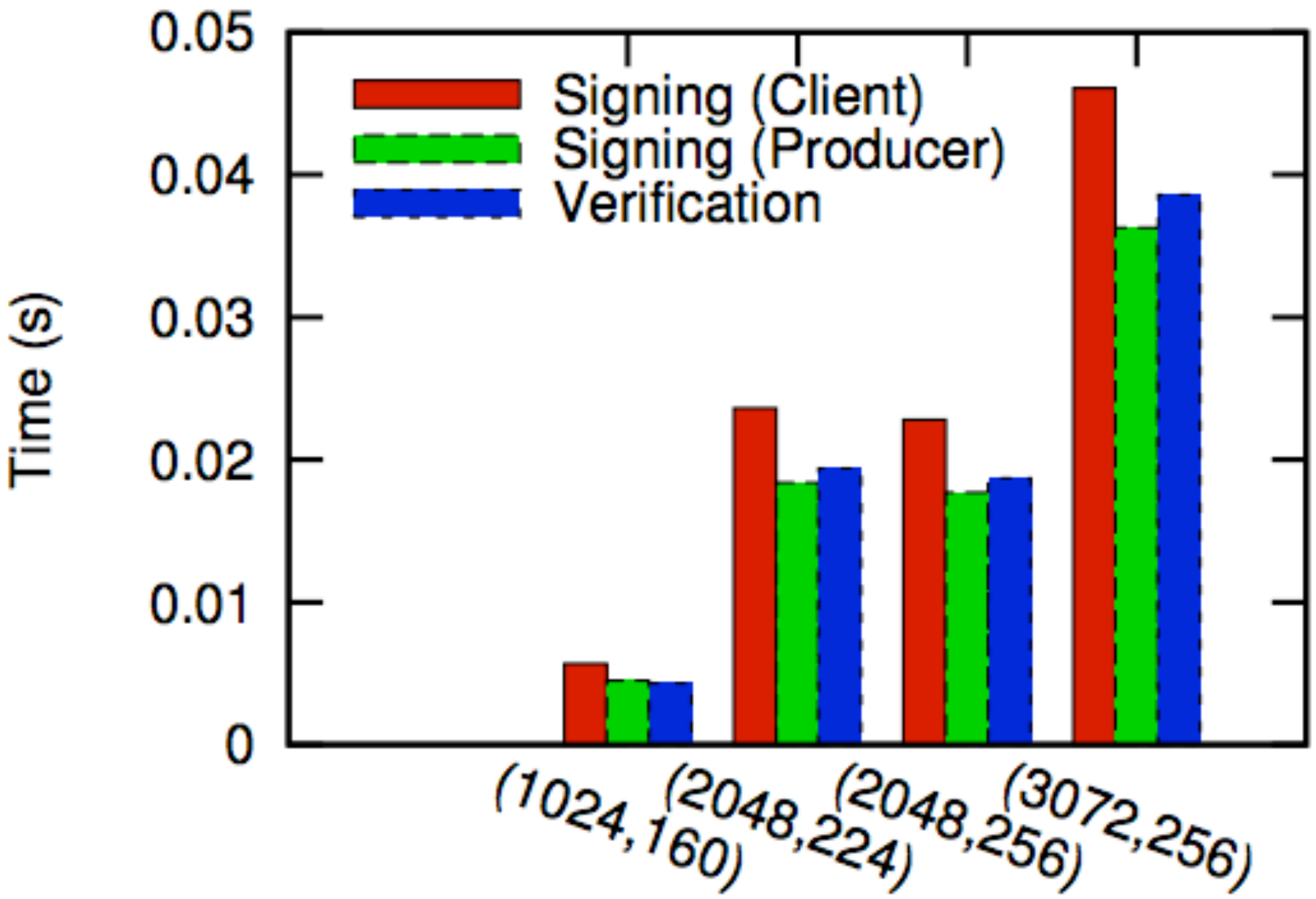


Key Parameters	Signature Size (Bytes)
(1024,160)	210
(2048,224)	287
(2048,256)	325
(3072,256)	326

- Network overhead between client and producer depends on the size (and hence strength) of the signature

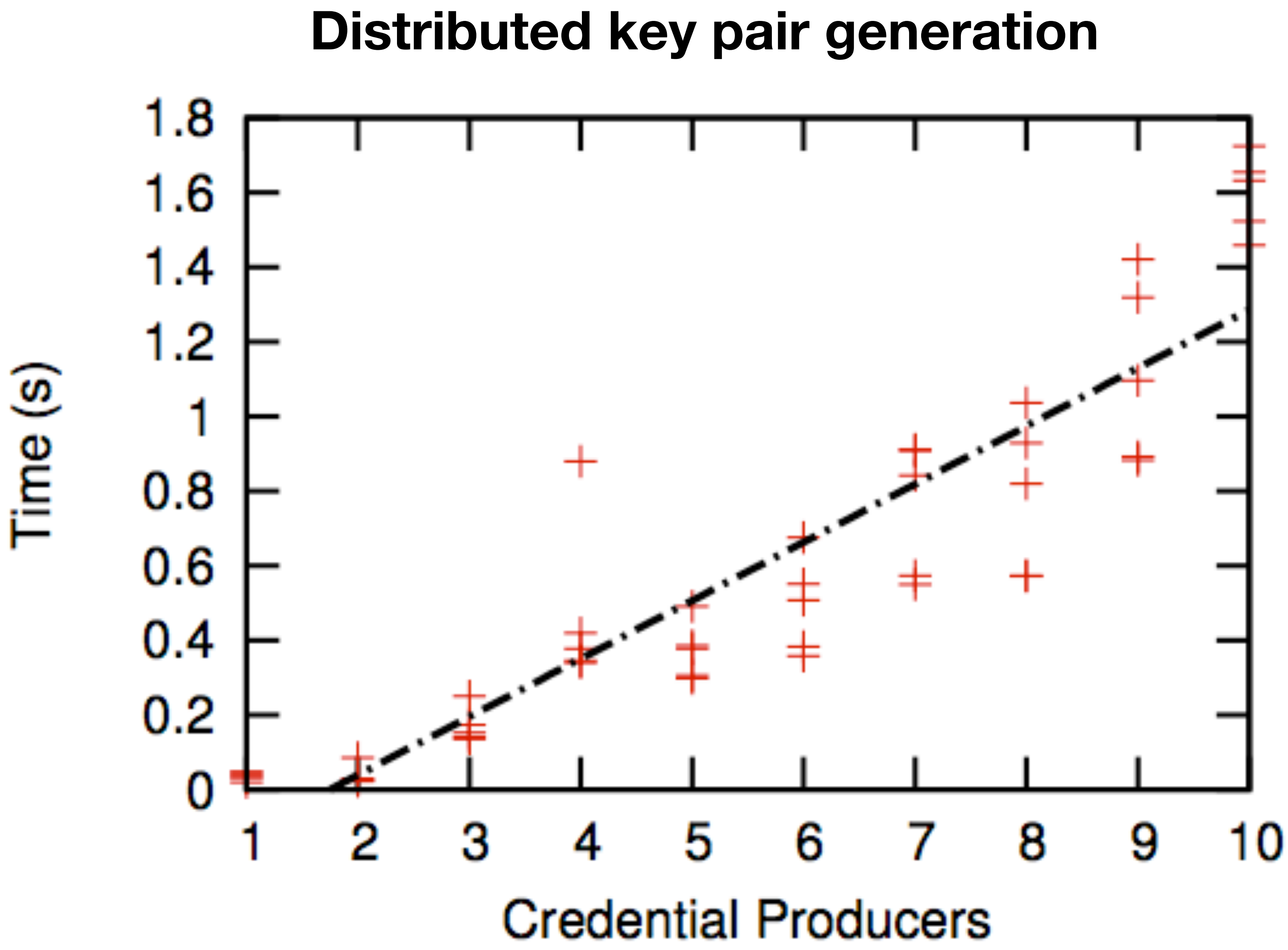
Evaluation: Producing/Consuming At-large Credentials

Blind Signature Operations



- For a 2048-bit signing key, credential production takes approximately 50ms of computation time, verification takes less than 20ms,

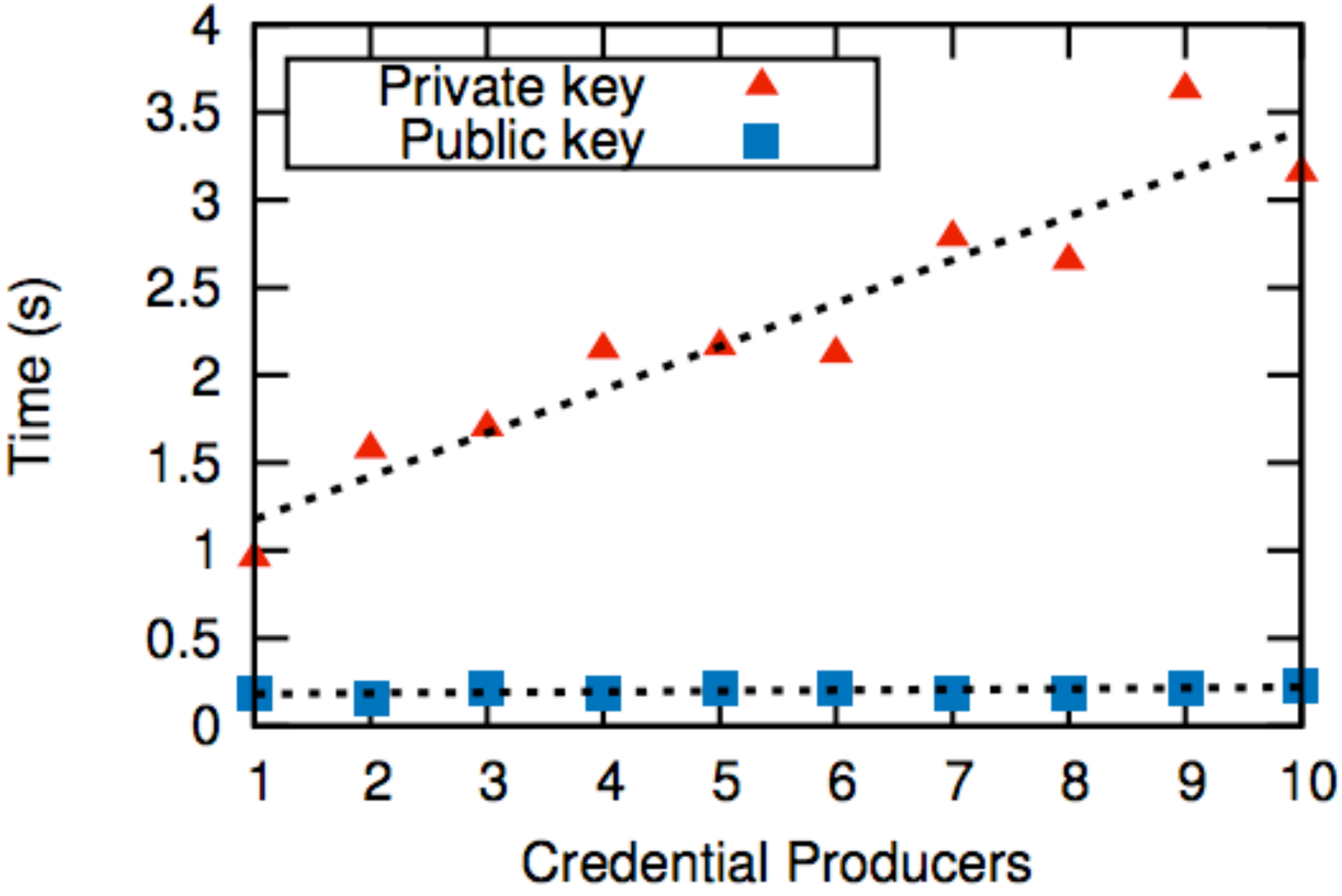
Evaluation: Producing Group Credentials



- **Key pair generation:** The first time a key pair is requested it is collectively generated by the producers

Evaluation: Producing Group Credentials

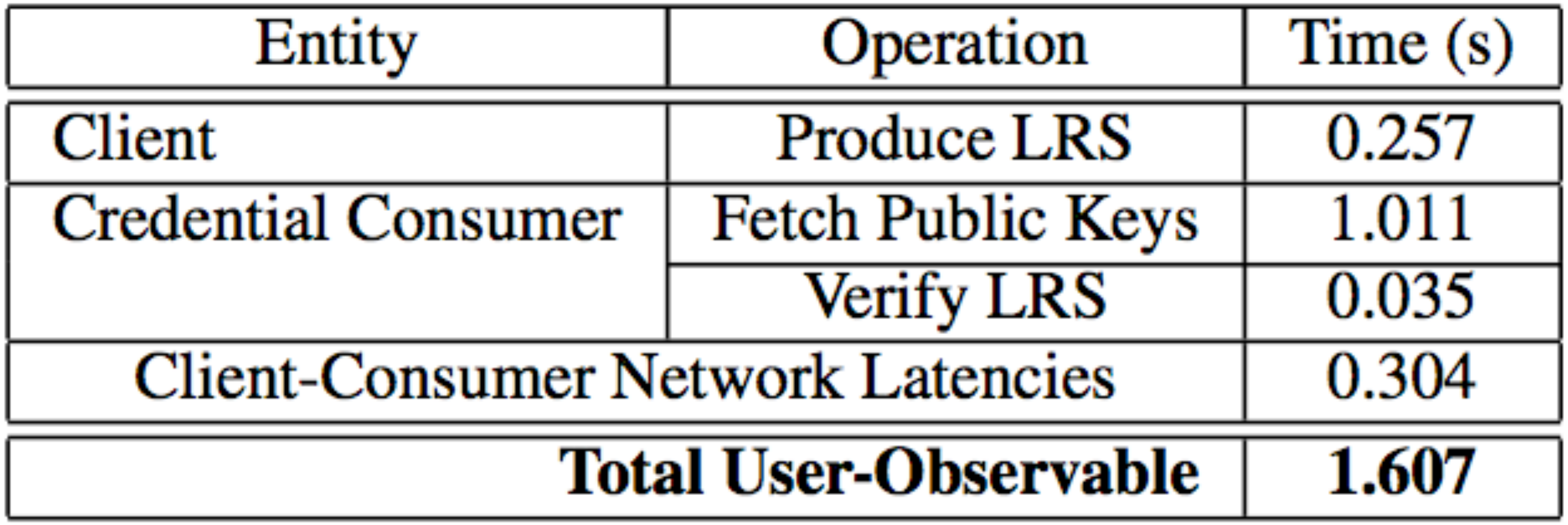
Retrieval of previously generated keys



- **Key retrieval:** requests to all producers are performed in parallel. Private keys include Facebook authentication

Evaluation: Consuming Credentials

End-to-end group credentials evaluation

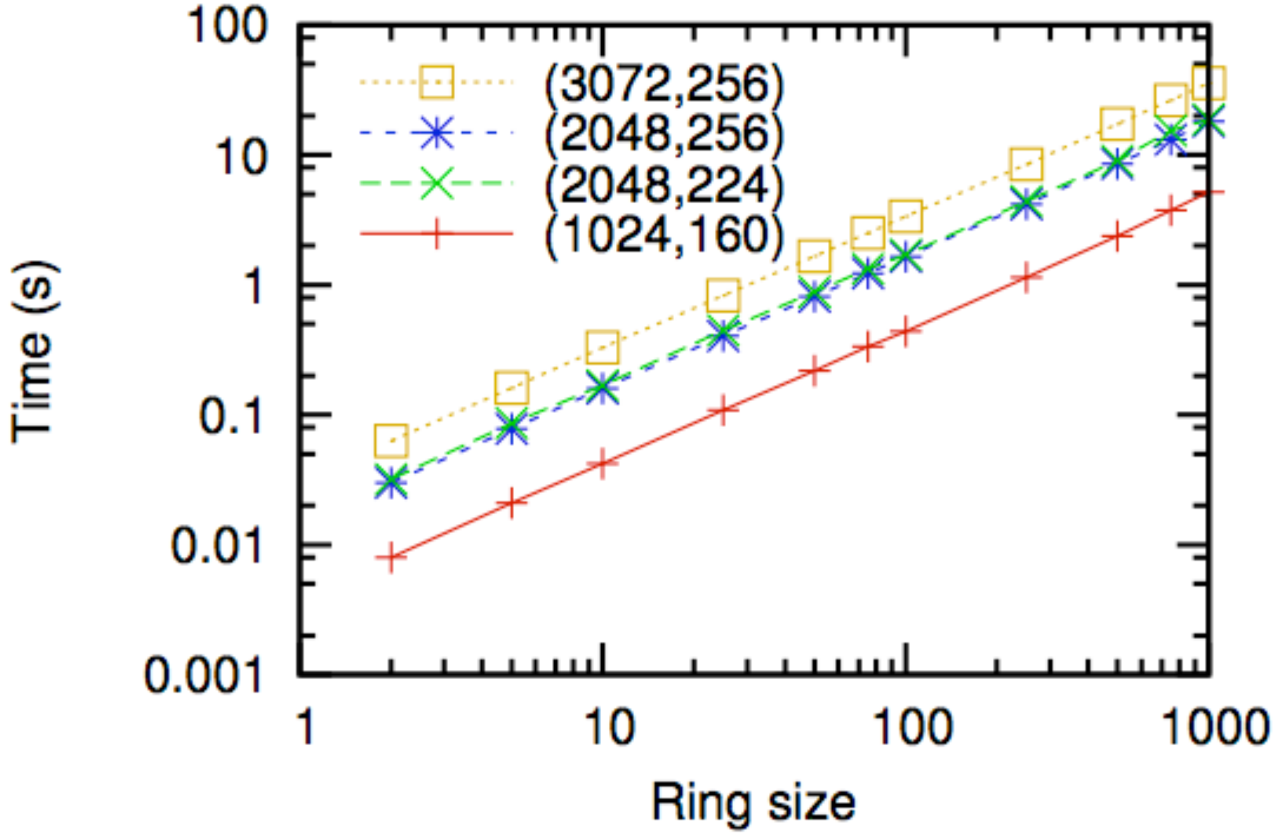


Entity	Operation	Time (s)
Client	Produce LRS	0.257
Credential Consumer	Fetch Public Keys	1.011
	Verify LRS	0.035
Client-Consumer Network Latencies		0.304
	Total User-Observable	**1.607**

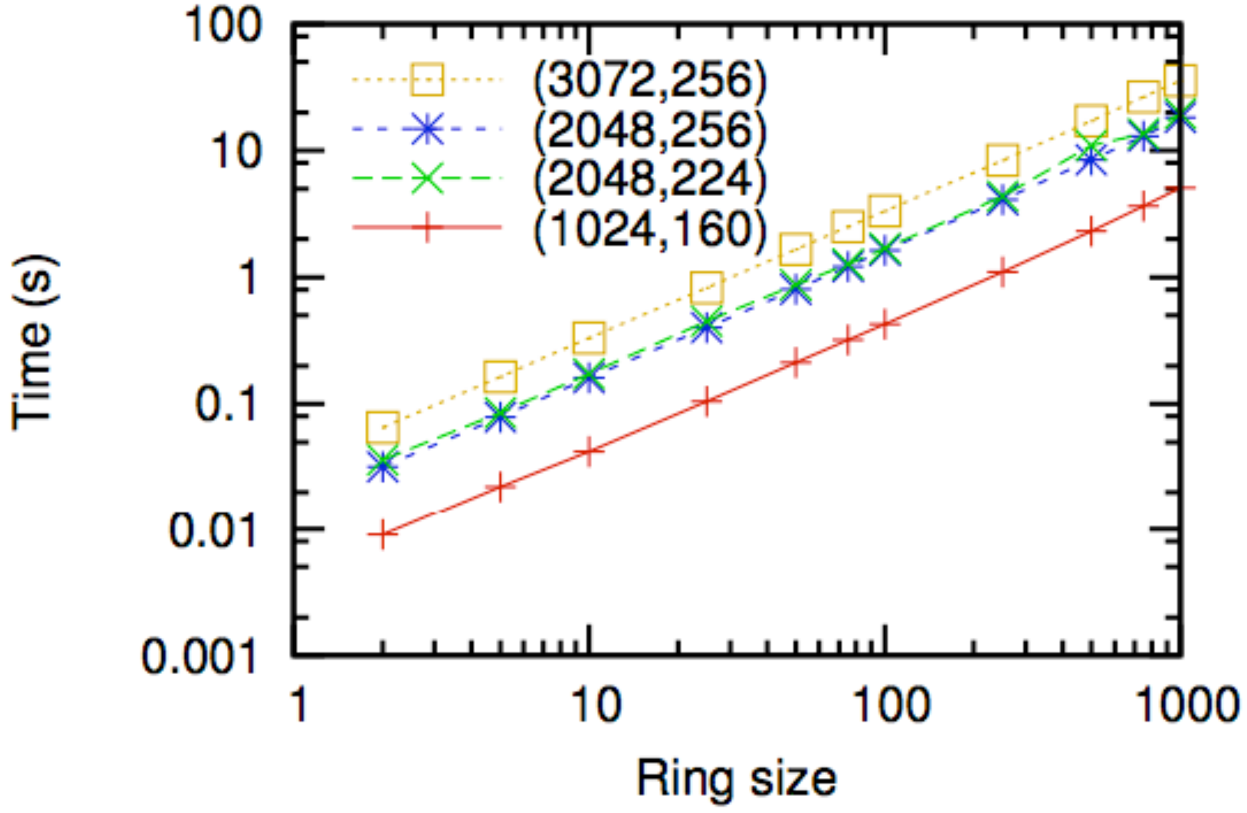
- **Group credential:** ten Facebook identities for DeDiS group
- 1.2s overhead vs non-anonymous federated authentication

Evaluation: Consuming Credentials

LRS signing



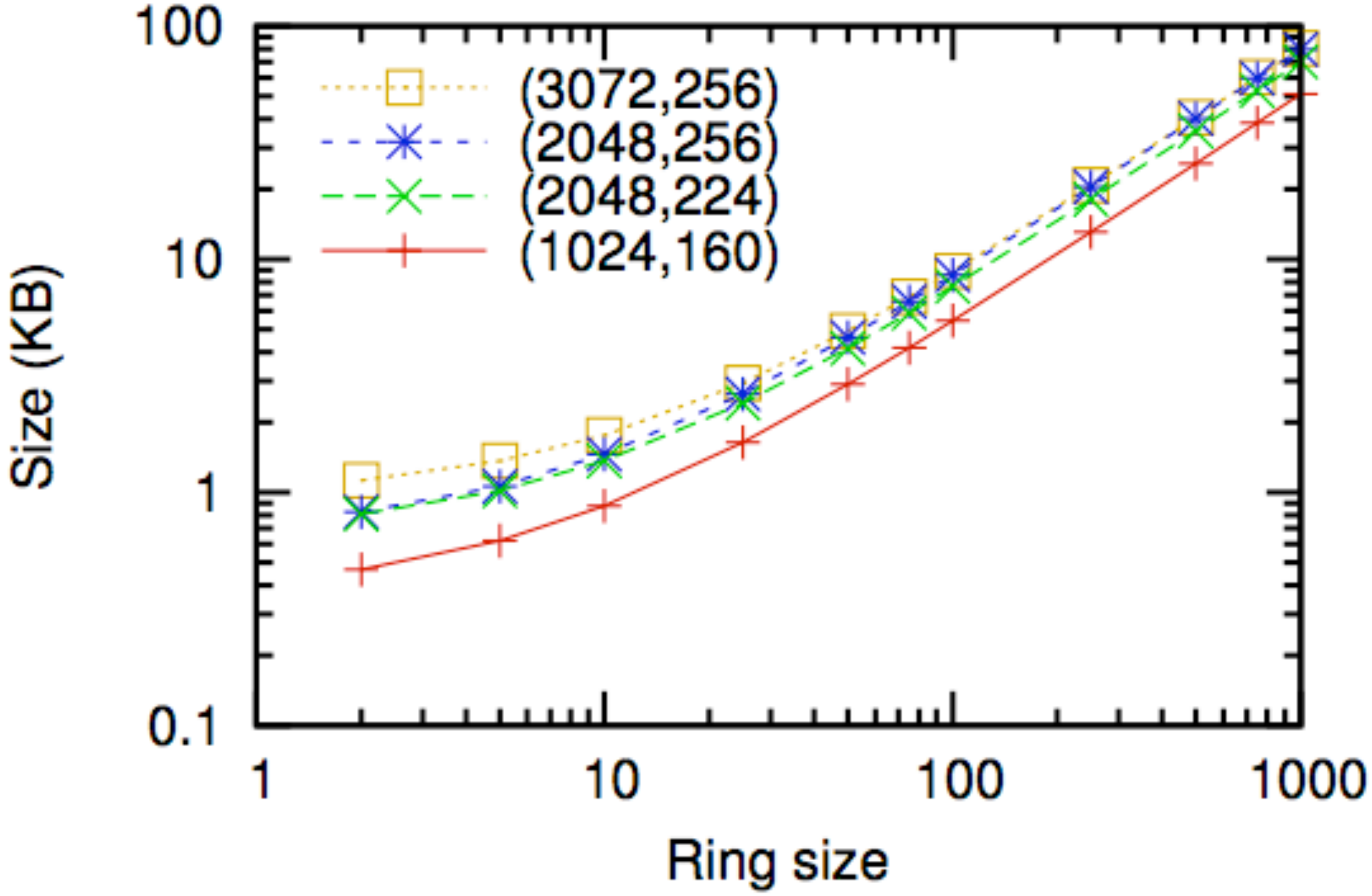
LRS verification



- For ring size ~100 (2048-bit keys), operations <1s

Evaluation: Consuming Credentials

LRS size (bandwidth)



- For ring sizes ~100 (2048-bit keys), signatures <10KB.

Roadmap

1. Background

2. Work Overview

3. System Architecture

4. Credential Producers and Consumers
   - At -Large Credentials
   - Group Credentials

5. Evaluation

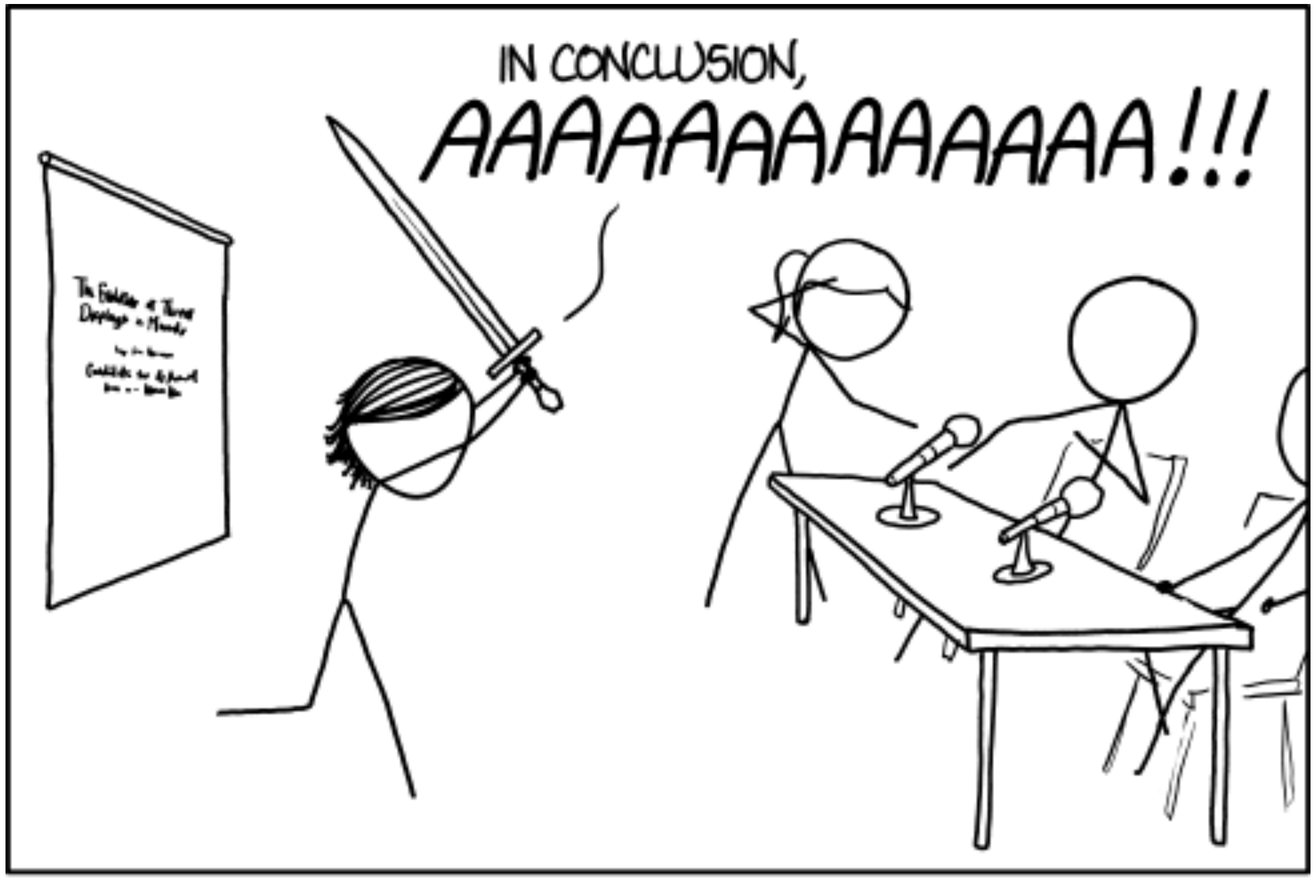
6. **Conclusions**

Conclusions and Future Directions

- Crypto-Book is a pluggable architecture for providing privacy preserving credentials based on federated identity providers.
- Experimental evaluations show acceptable overheads
- Privacy conscious applications can be developed on top of this platform
- Pluggable nature means other privacy preserving technologies can be integrated in future

Acknowledgements

- Thanks to my adviser, Bryan Ford and committee members, Joan Feigenbaum, Ramki Gummadi, Anil Somayaji
- Collaborators: Danny Jackowitz, Ennan Zhai, David Isaac Wolinsky and DeDiS research group members Ewa Syta, Weiyi Wu and Jose Faleiro
- Undergraduate adviser: The late Robin Milner (University of Cambridge, UK)
- PhD funding sources: Yale University, NSF grant CCF-0916389, DARPA SAFER grant N66001-11-C-4018
- Thanks to the everyone in the Yale Computer Science department and everyone else for attending



THE BEST THESIS DEFENSE IS A GOOD THESIS OFFENSE.

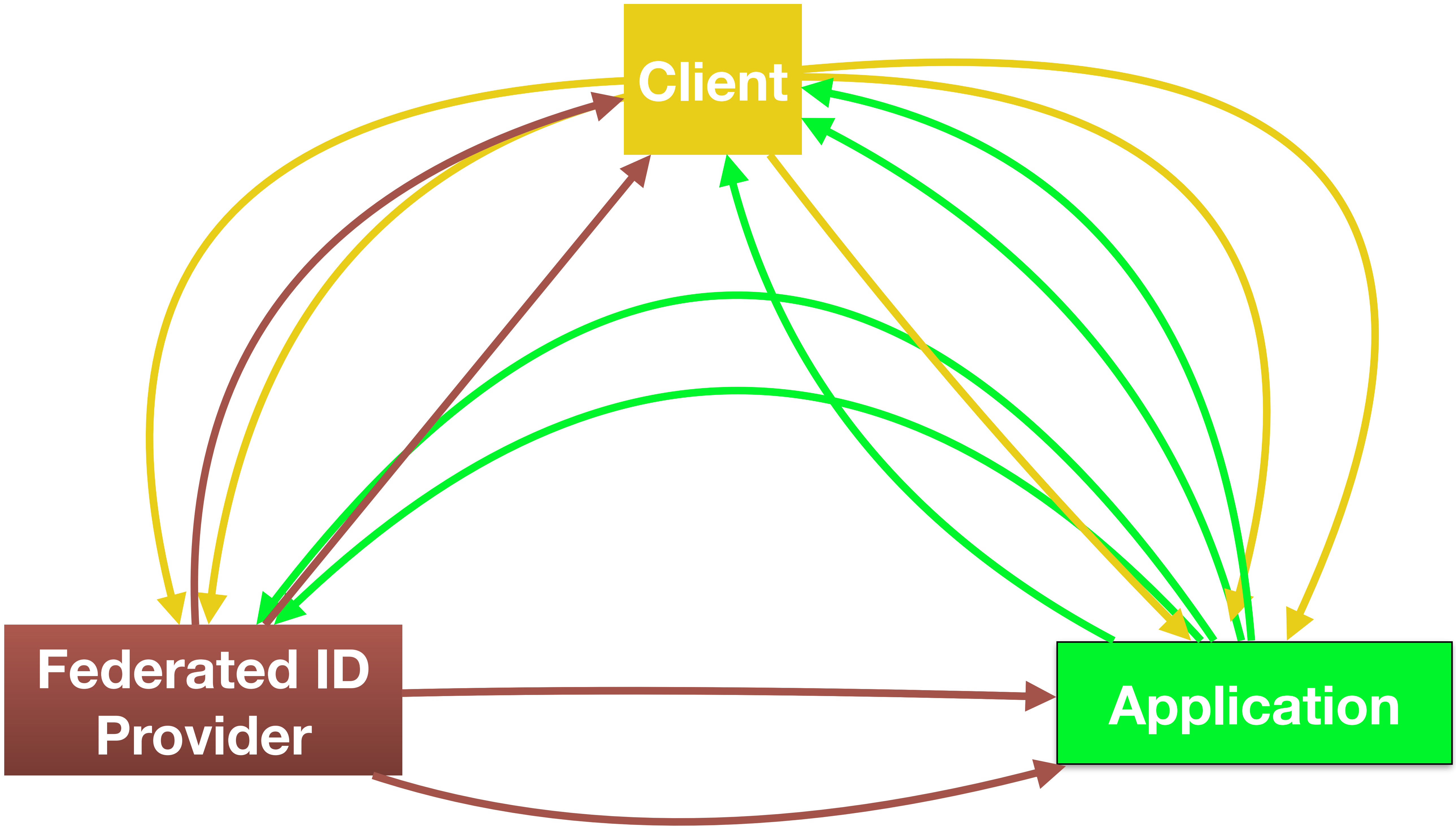
Thanks!

[Subsequent slides are were removed from presentation and may be incomplete]

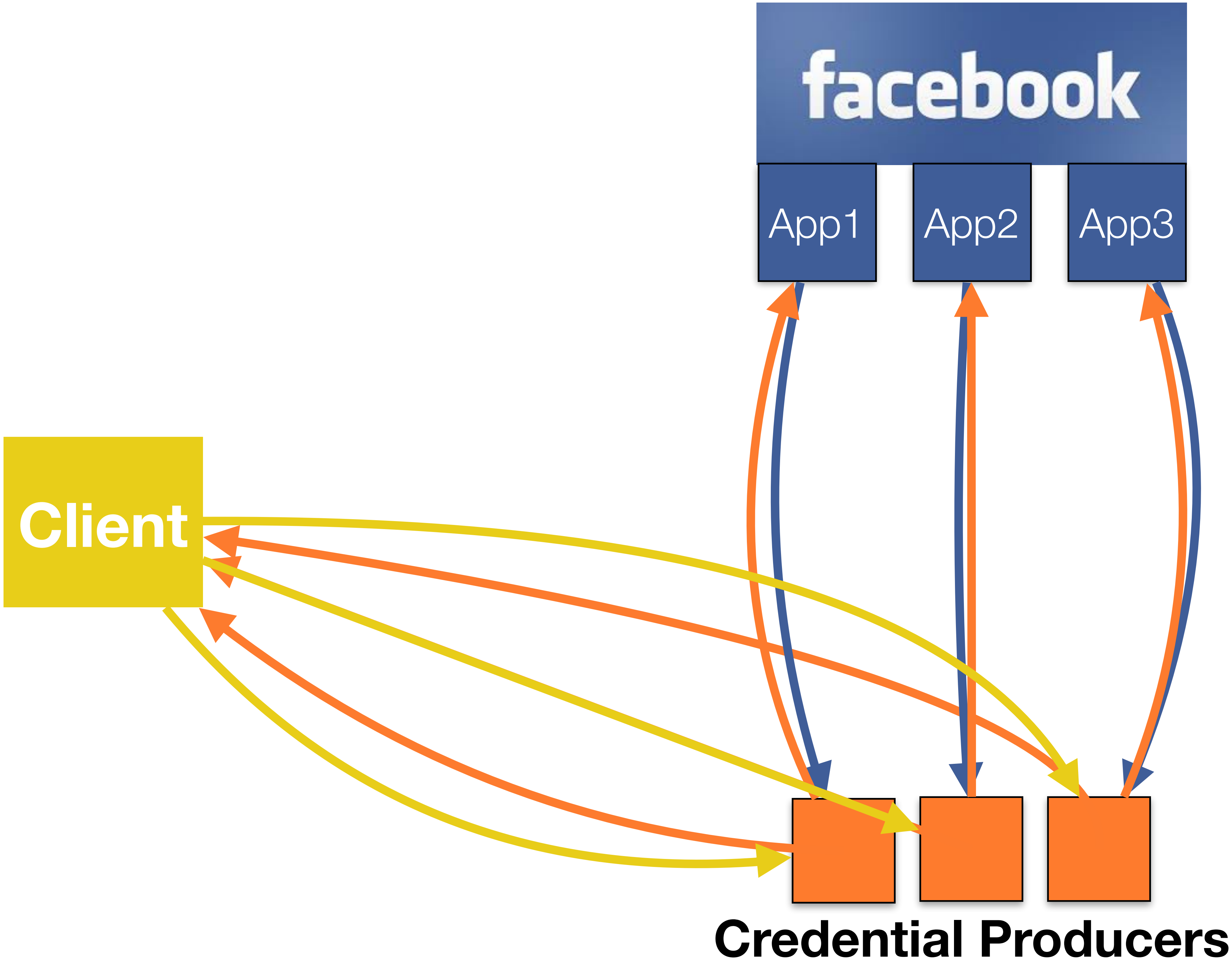
“Two Principles of Deadlines:
1. All deadlines converge on the same day—
Deadline Day.
2. Every day is Deadline Day.”

–Bryan Ford

Federated Authentication Interaction



Credential Assignment Mechanism



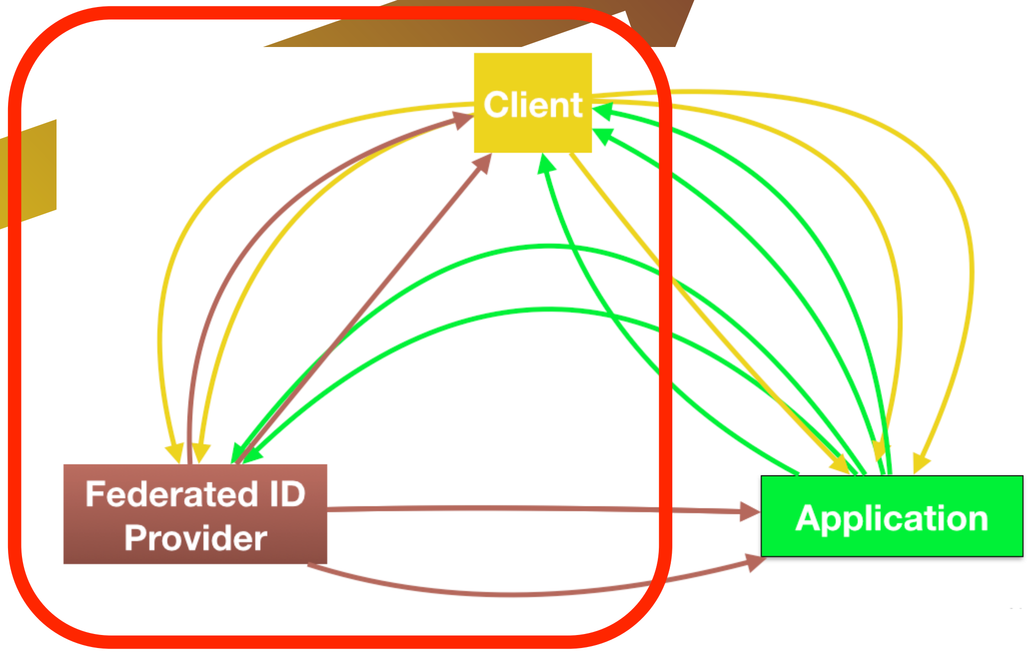
Credential Assignment Mechanism

Federated ID Provider(s)

PayPal™

facebook

Client

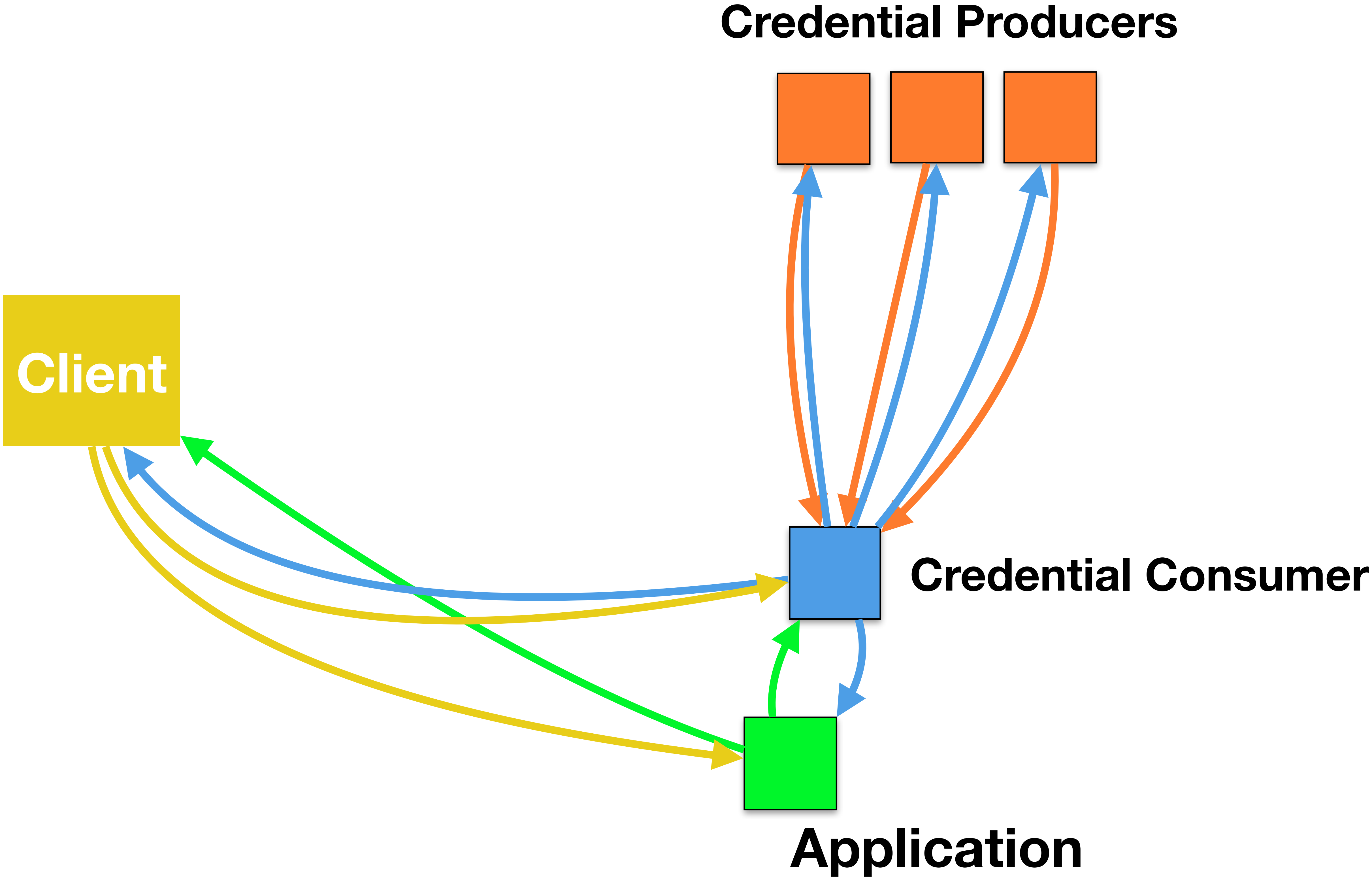


PayPal™

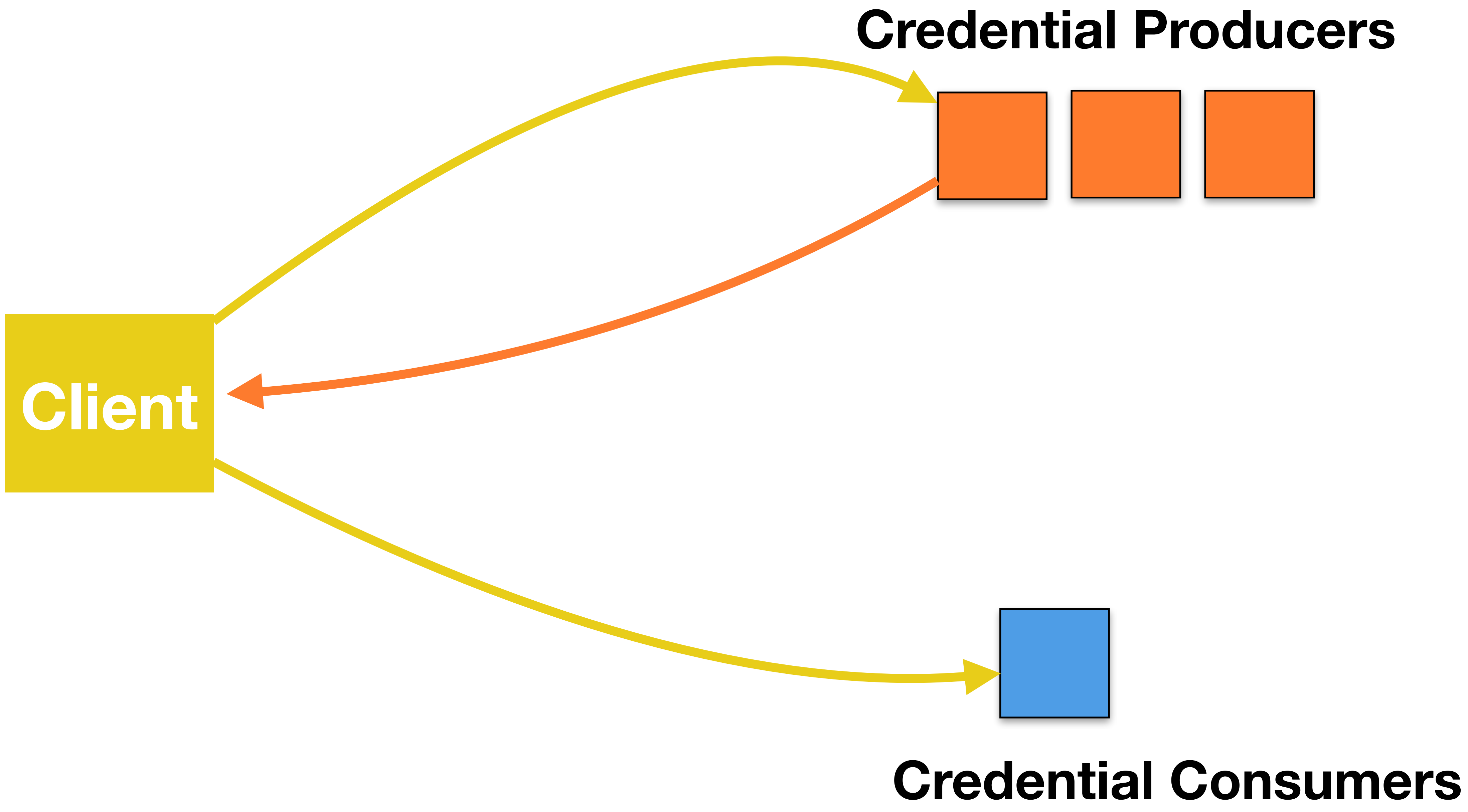
App1 App2 App3

Credential Producers

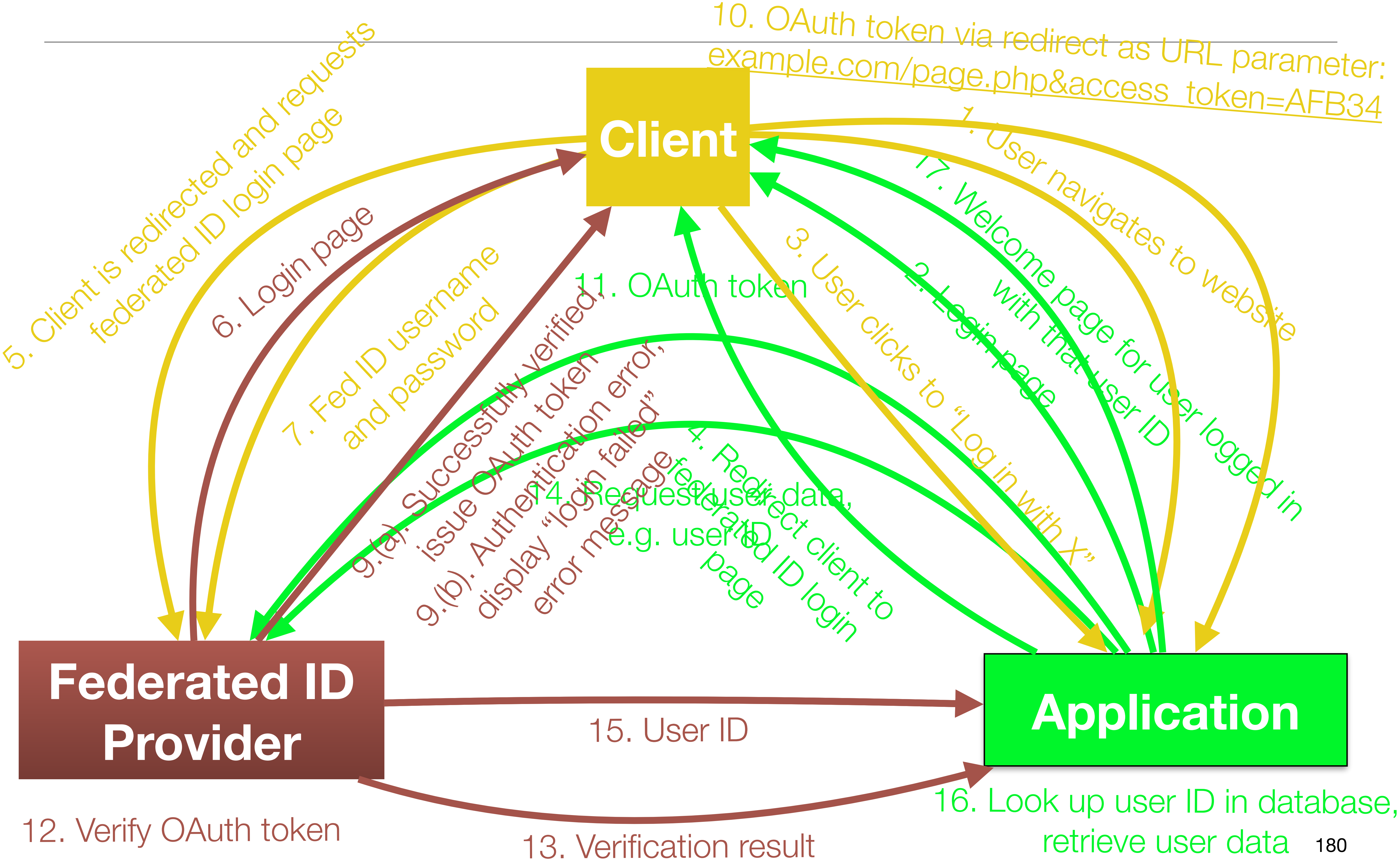
System Architecture



At-Large Credential Scheme



Federated Authentication Interaction



Client

Federated ID Provider

Application

1. User navigates to website

2. Login page

3. User clicks to “Log in with X”

4. Redirect client to federated ID login page

5. Client is redirected and requests federated ID login page

6. Login page

7. Fed ID username and password

9.(a). Successfully verified, issue OAuth token

9.(b). Authentication error, display “login failed” error message

10. OAuth token via redirect as URL parameter: example.com/page.php&access_token=AFB34

11. OAuth token

12. Verify OAuth token

13. Verification result

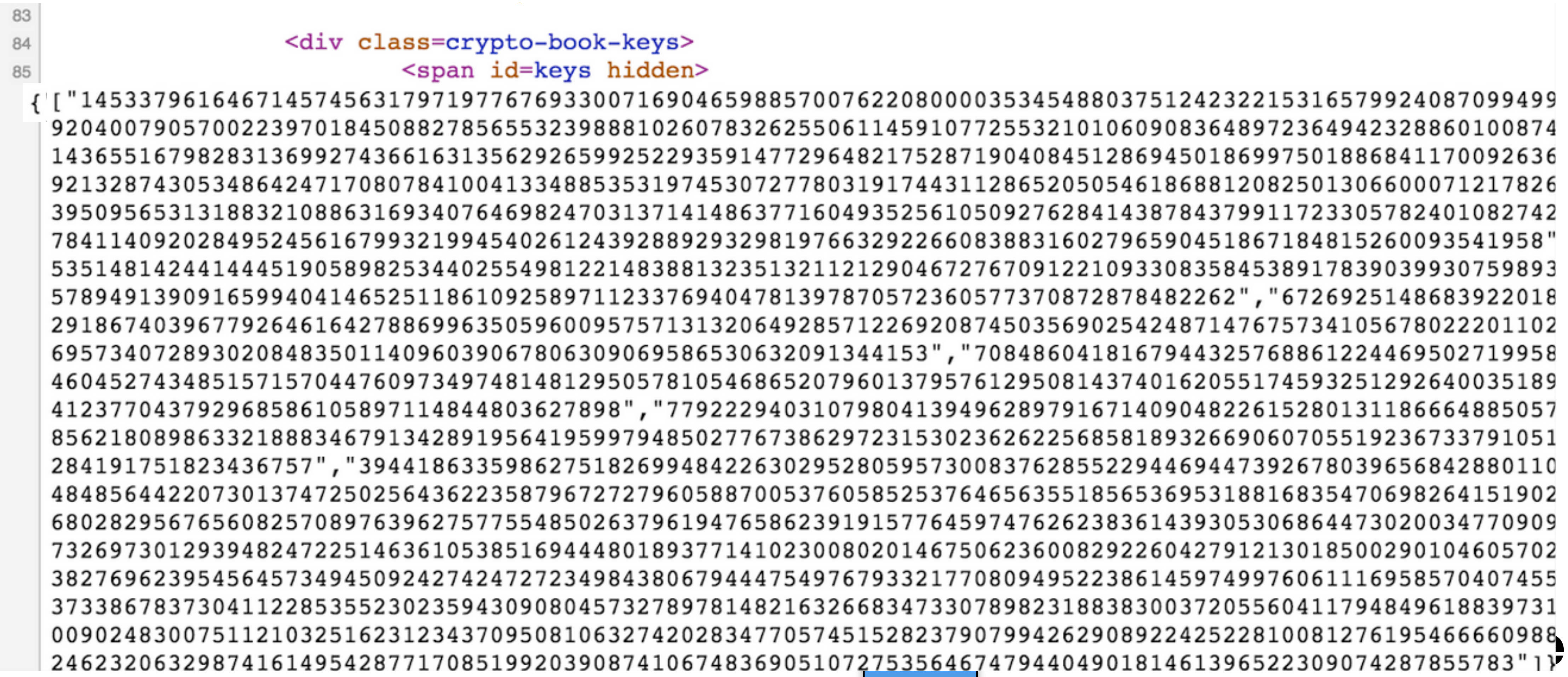
14. Request user data, e.g. user ID

15. User ID

16. Look up user ID in database, retrieve user data

17. Welcome page for user logged in with that user ID

Group Credential Scheme



```

                        <div class=crypto-book-keys>
                                <span id=keys hidden>
{'["14533796164671457456317971977676933007169046598857007622080000353454880375124232215316579924087099499
92040079057002239701845088278565532398881026078326255061145910772553210106090836489723649423288601008 74
1436551679828313699274366163135629265992522935914772964821752871904084512869450186997501886841170092636
9213287430534864247170807841004133488535319745307277803191744311286520505461868812082501306600071217826
3950956531318832108863169340764698247031371414863771604935256105092762841438784379911723305782401082742
7841140920284952456167993219945402612439288929329819766329226608388316027965904518671848152600935419 58"
5351481424414445190589825344025549812214838813235132112129046727670912210933083584538917839039930759893
578949139091659940414652511861092589711233769404781397870572360577370872878482262","6726925148683922018
2918674039677926461642788699635059600957571313206492857122692087450356902542487147675734105678022201102
695734072893020848350114096039067806309069586530632091344153","7084860418167944325768861224469502719958
4604527434851571570447609734974814812950578105468652079601379576129508143740162055174593251292640035189
412377043792968586105897114844803627898","7792229403107980413949628979167140904822615280131186664885057
8562180898633218883467913428919564195997948502776738629723153023626225685818932669060705519236733791051
284191751823436757","3944186335986275182699484226302952805957300837628552294469447392678039656842880110
4848564422073013747250256436223587967272796058870053760585253764656355185653695318816835470698264151902
6802829567656082570897639627577554850263796194765862391915776459747626238361439305306864473020034770909
7326973012939482472251463610538516944480189377141023008020146750623600829226042791213018500290104605702
3827696239545645734945092427424727234984380679444754976793321770809495223861459749976061116958570407455
3733867837304112285355230235943090804573278978148216326683473307898231888383003720556041179484961883 9731
0090248300751121032516231234370950810632742028347705745152823790799426290892242522810081276195466660988
2462320632987416149542877170851992039087410674836905107275356467479440490181461396522309074287855783"]}
```

Clie

er

Chat Room Application