Integrating DAGA into the cothority framework and using it to build a login service

DEDIS, EPFL 2018/19 - Lucas Pires

Responsible: Prof. Bryan Ford, Dr. Ewa Syta

Supervisor: Linus Gasser

Integrating DAGA into the cothority framework and using it to build a login service

Deniable **A**nonymous **G**roup **A**uthentication

- Decentralized Authentication Protocol
- Forward-security, etc. more later

Motivation / Intro

- Authentication Identification and Privacy
- ➔ where possible, get rid of identification
- ➔ DAGA
- GOAL: offer easy way to use DAGA, Login Service

Overview

- Background / DAGA
- Cothority implementation
- Authentication delegation
- PoC & demo
- Conclusion

Background / DAGA

Background / DAGA –

DAGA

Background / DAGA –

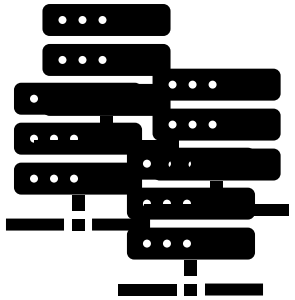
Entity / user

DAGA

Background / DAGA –

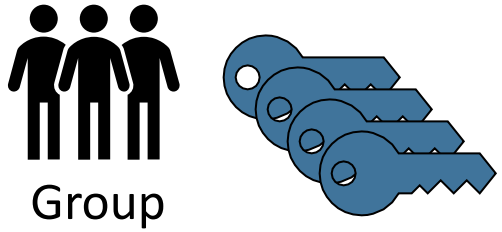
Entity / user

DAGA

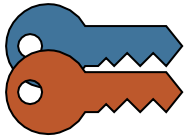


Anytrust servers

Background / DAGA –



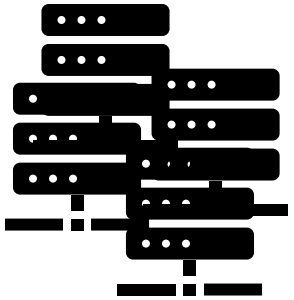
Entity / user



DAGA

Auth. request

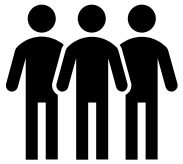
Decision



Anytrust servers

Background / DAGA –

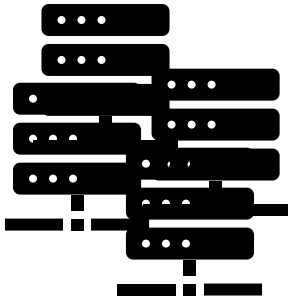
- Completeness
- Soundness



Group

Entity / user

DAGA



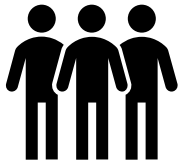
Anytrust servers

Auth. request

Decision

Background / DAGA –

- Completeness
- Soundness
- Anonymity



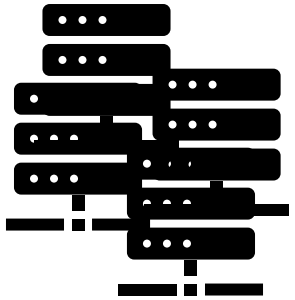
Group

DAGA

Entity / user

Auth. request

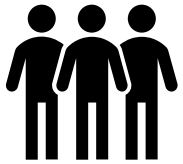
Decision



Anytrust servers

Background / DAGA –

- Anonymity
- Proportionality



Group

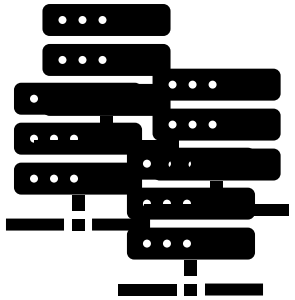
Entity / user

DAGA

Auth. request

Decision

+ Linkage Tag



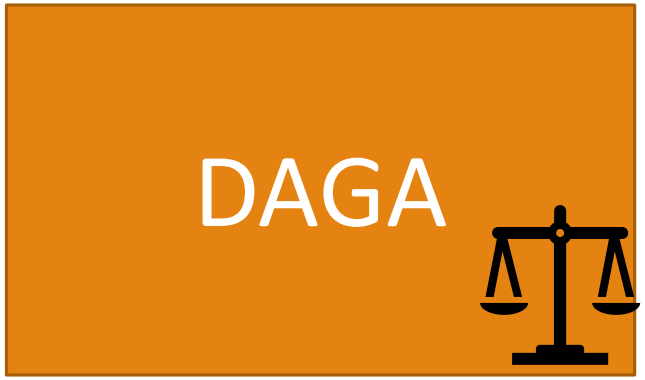
Anytrust servers

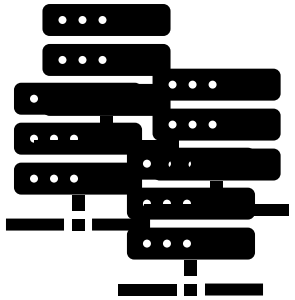
Background / DAGA –

- Anonymity
- Proportionality
- Deniability

Group

Entity / user





Anytrust servers

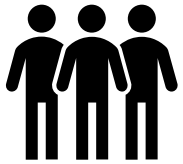
Auth. request

Decision

+ Linkage Tag

Background / DAGA –

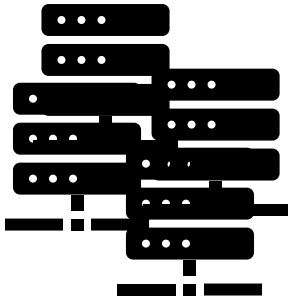
- Anonymity
- Proportionality
- Deniability
- Forward security



Group

Entity / user

DAGA



Anytrust servers

Auth. request

Decision

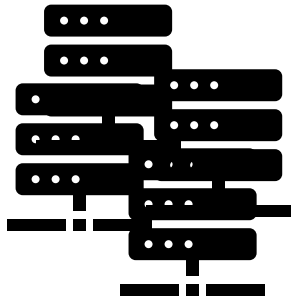
+ Linkage Tag

Background / DAGA –

Prover

Build request / client's protocol

Context



Verifiers

Adapted / redrawn from https://github.com/dedis/student_17/blob/master/pfs_pop/presentation_pfs_pop.pdf

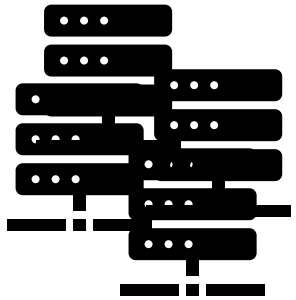
Background / DAGA –

Prover

Build request / client's protocol

Initial tag

Context



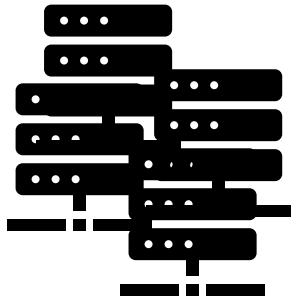
Verifiers

Adapted / redrawn from https://github.com/dedis/student_17/blob/master/pfs_pop/presentation_pfs_pop.pdf

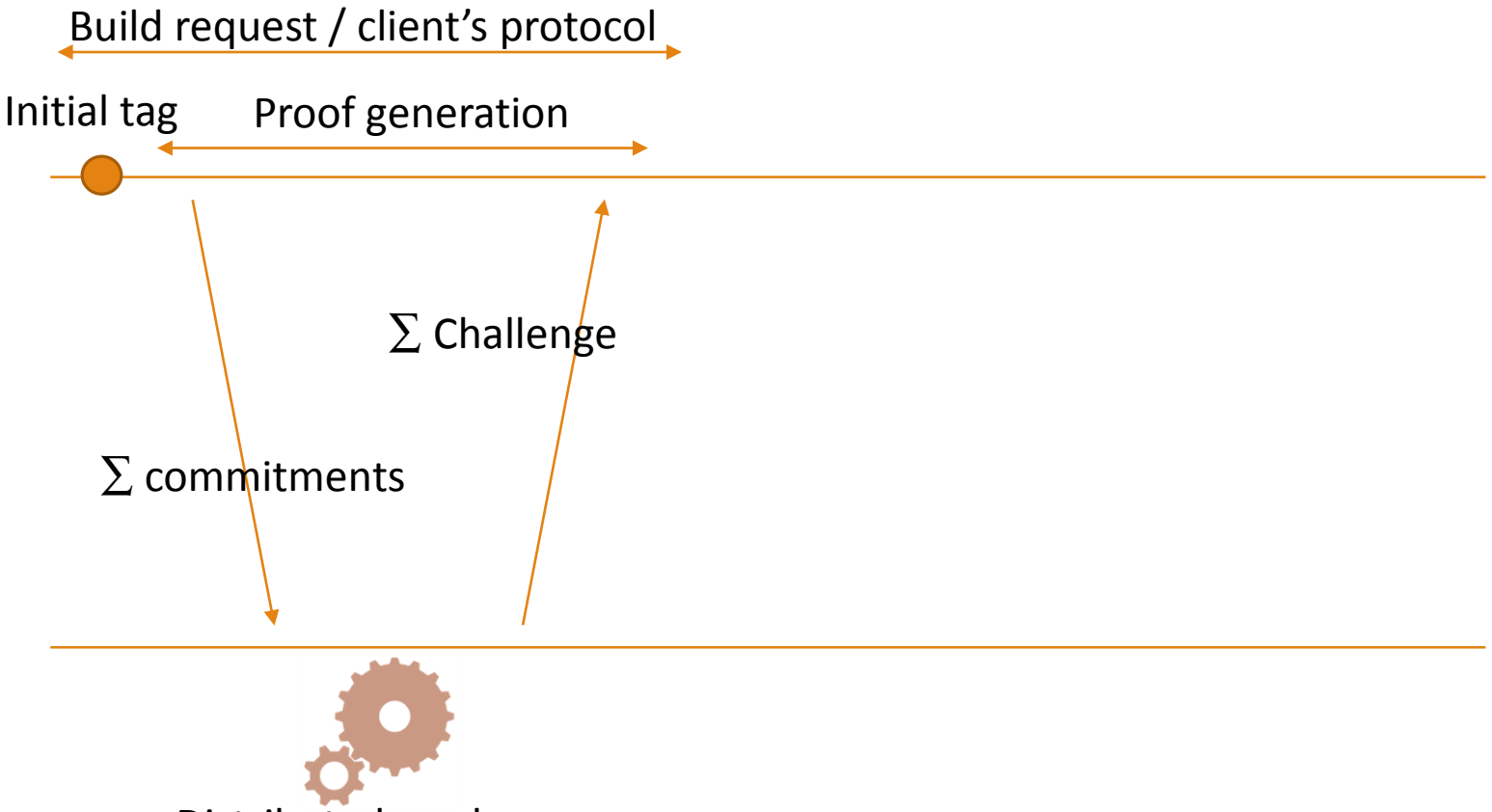
Background / DAGA –

Prover

Context



Verifiers

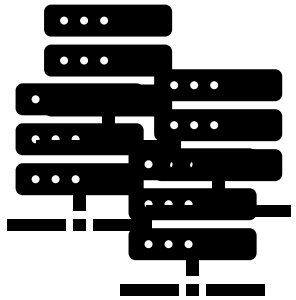


Adapted / redrawn from https://github.com/dedis/student_17/blob/master/pfs_pop/presentation_pfs_pop.pdf

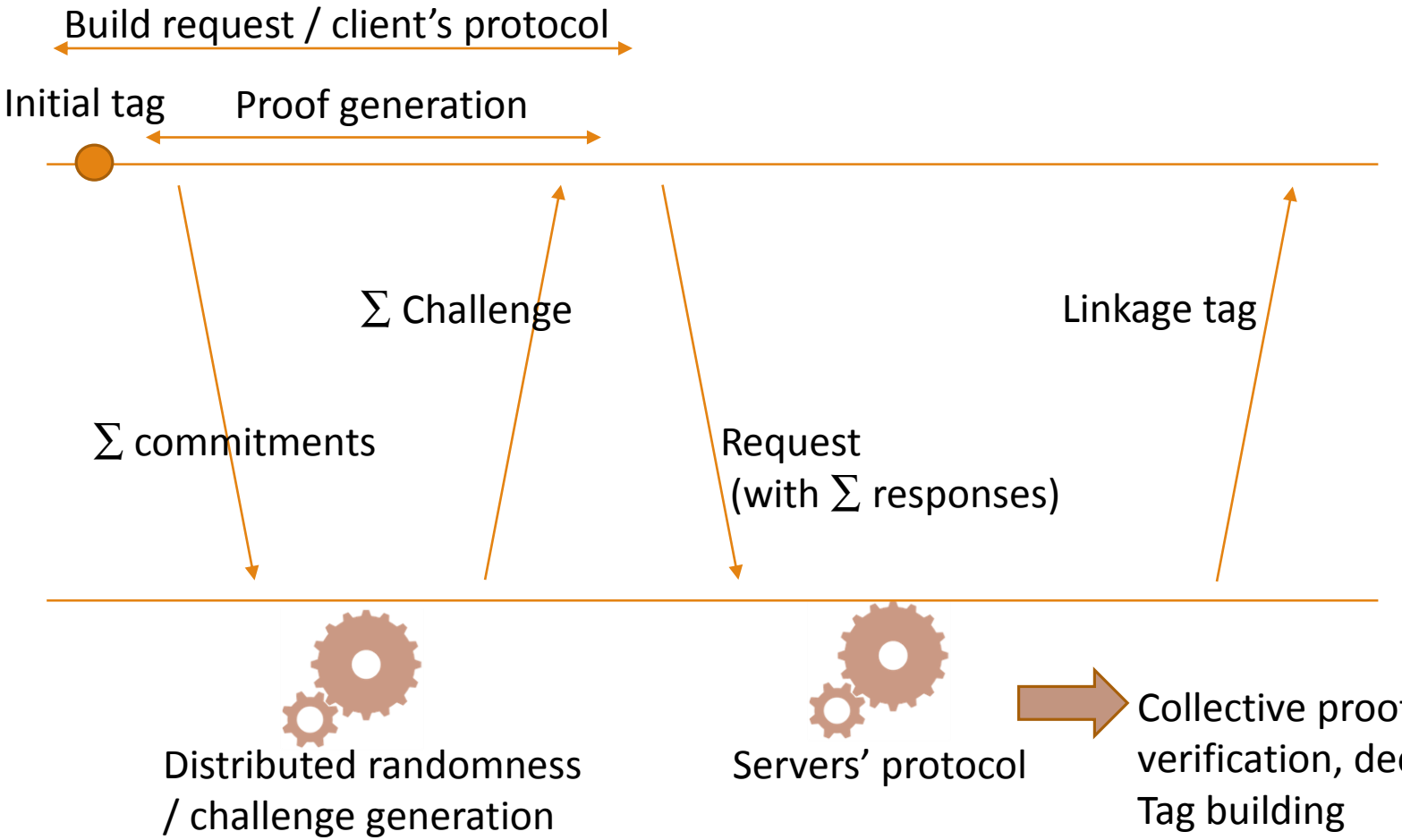
Background / DAGA –

Prover

Context



Verifiers



Adapted / redrawn from https://github.com/dedis/student_17/blob/master/pfs_pop/presentation_pfs_pop.pdf

Overview

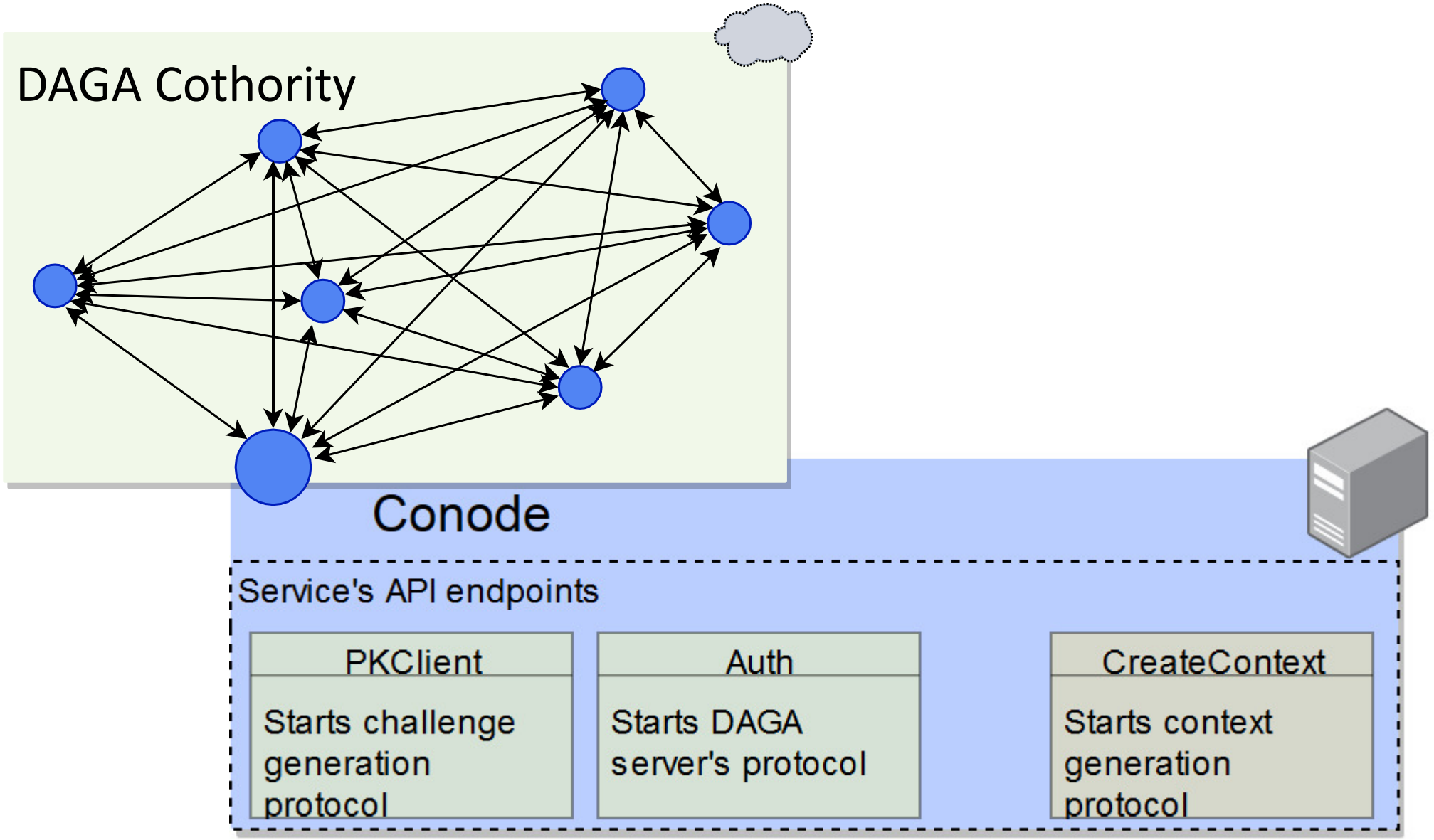
- Background / DAGA
- Cothority implementation
- Authentication delegation
- PoC demo
- Conclusion &? Future

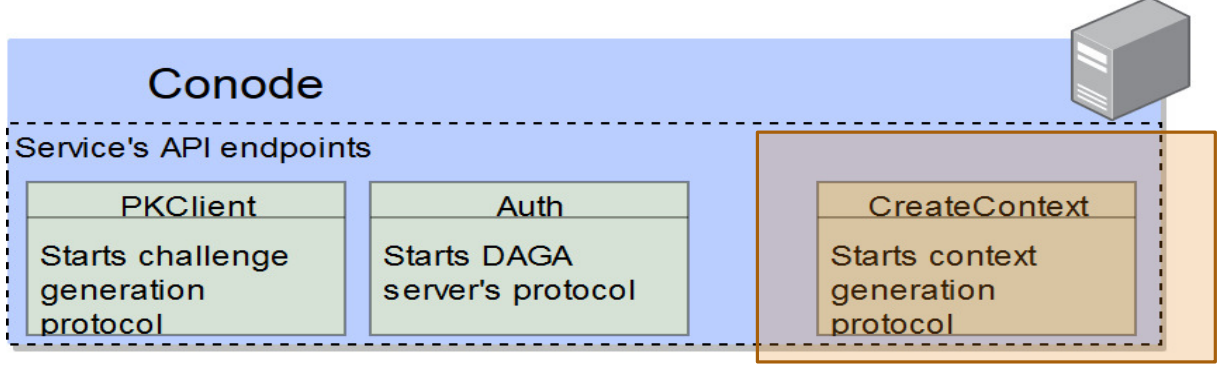
Cothority Implementation

- DAGA Library (continuation of A. Villard's work)
- New *Service & Protocols*
  (context generation / challenge generation / DAGA servers' protocol)
- Can run simulations locally and on DETERLab
- 80% code coverage
- Possible to generate proto files
- CLI client

Cothority Implementation

- DAGA Library (continuation of A. Villard's work)
- New *Service & Protocols*
  (context generation / challenge generation / DAGA servers' protocol)
- Can run simulations locally and on DETERLab
- 80% code coverage
- Possible to generate proto files
- CLI client

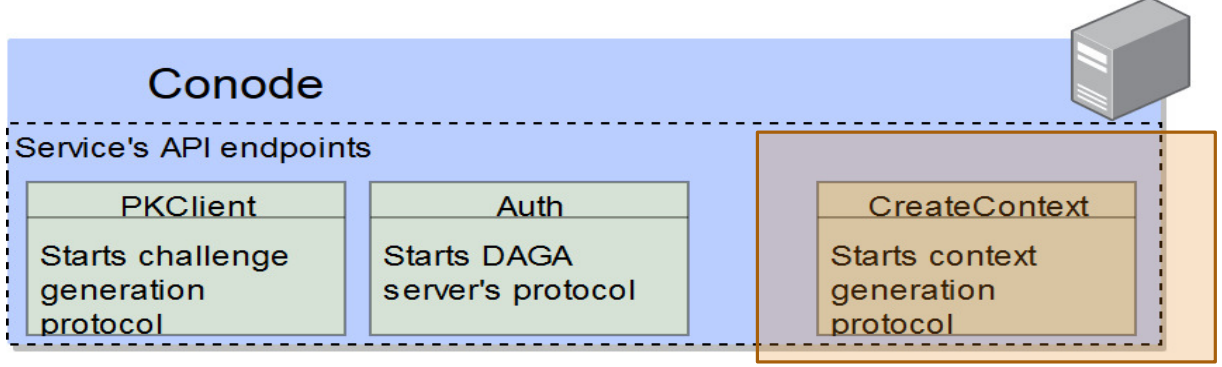

DAGA Cothority
Conode
Service's API endpoints
PKClient
Starts challenge generation protocol
Auth
Starts DAGA server's protocol
CreateContext
Starts context generation protocol



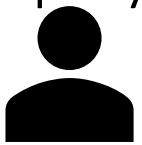
Client /
3rd party service admin

Administrative phase

1) Collect public **keys** of subscribers
2) Build a **roster** of willing conodes (partnerships or open access nodes)



Client /
3rd party service admin

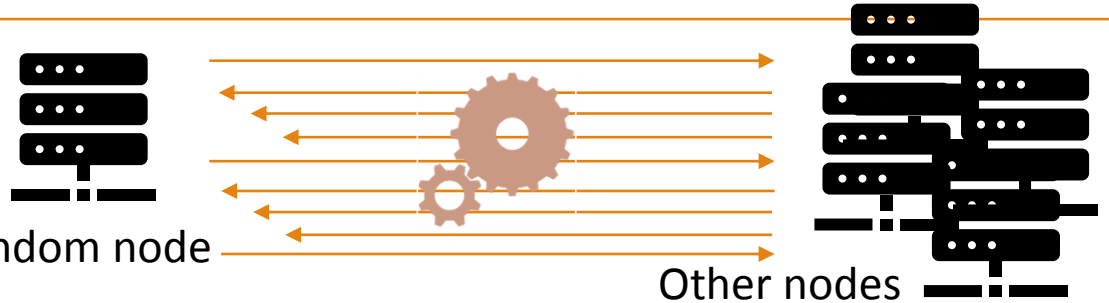


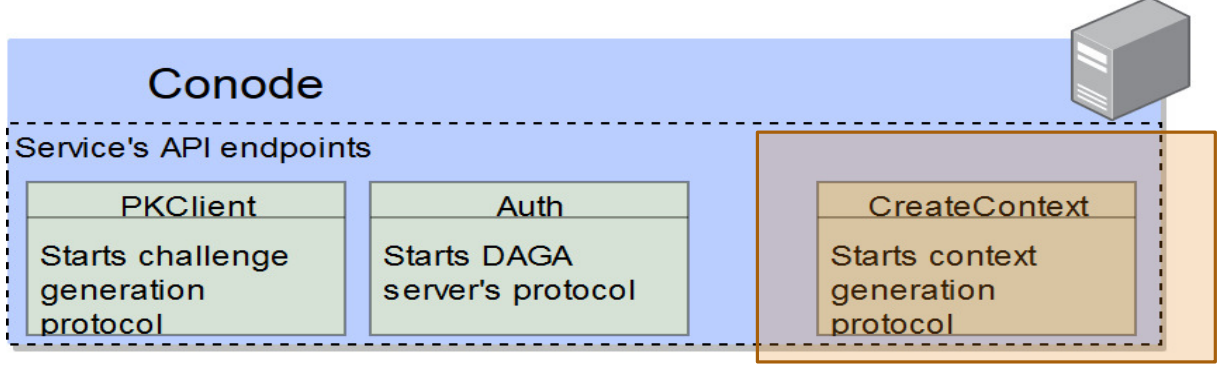
Administrative phase

1) Collect public **keys** of subscribers
2) Build a **roster** of willing conodes (partnerships or open access nodes)

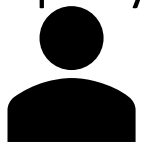
3) Call CreateContext(**keys**, **roster**)

Context generation protocol





Client /
3rd party service admin



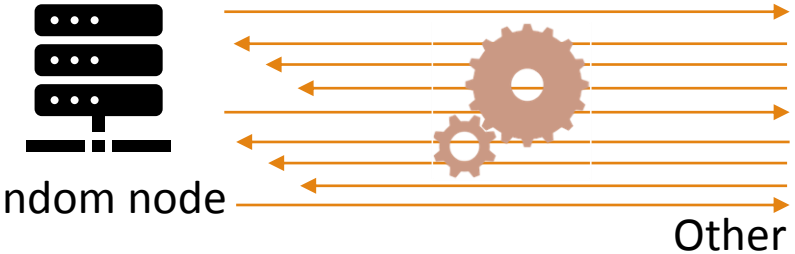
Administrative phase

1) Collect public **keys** of subscribers
2) Build a **roster** of willing conodes (partnerships or open access nodes)

3) Call CreateContext(**keys**, **roster**)

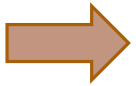
Context

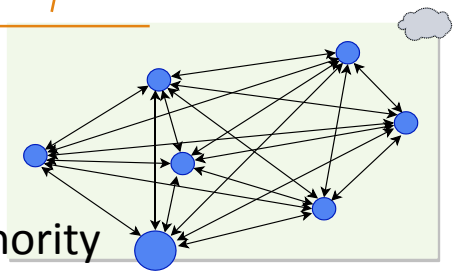
Context generation protocol



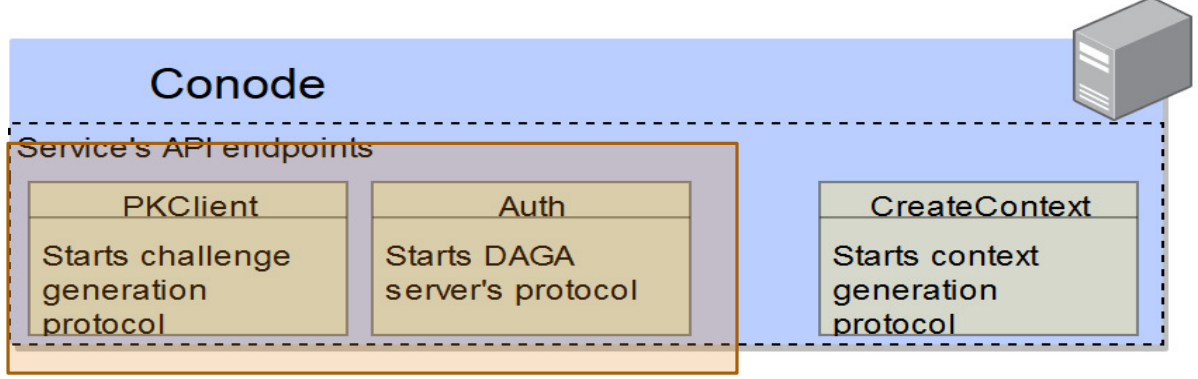
Random node

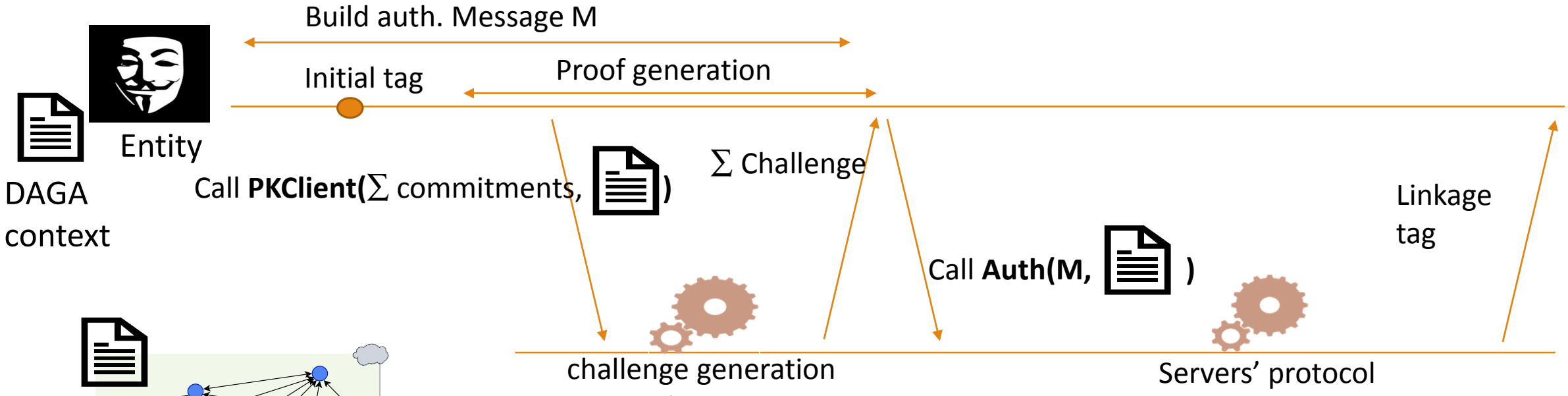
Other nodes

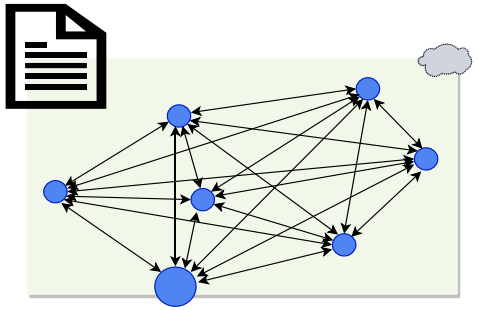




New Cothority
For the new context





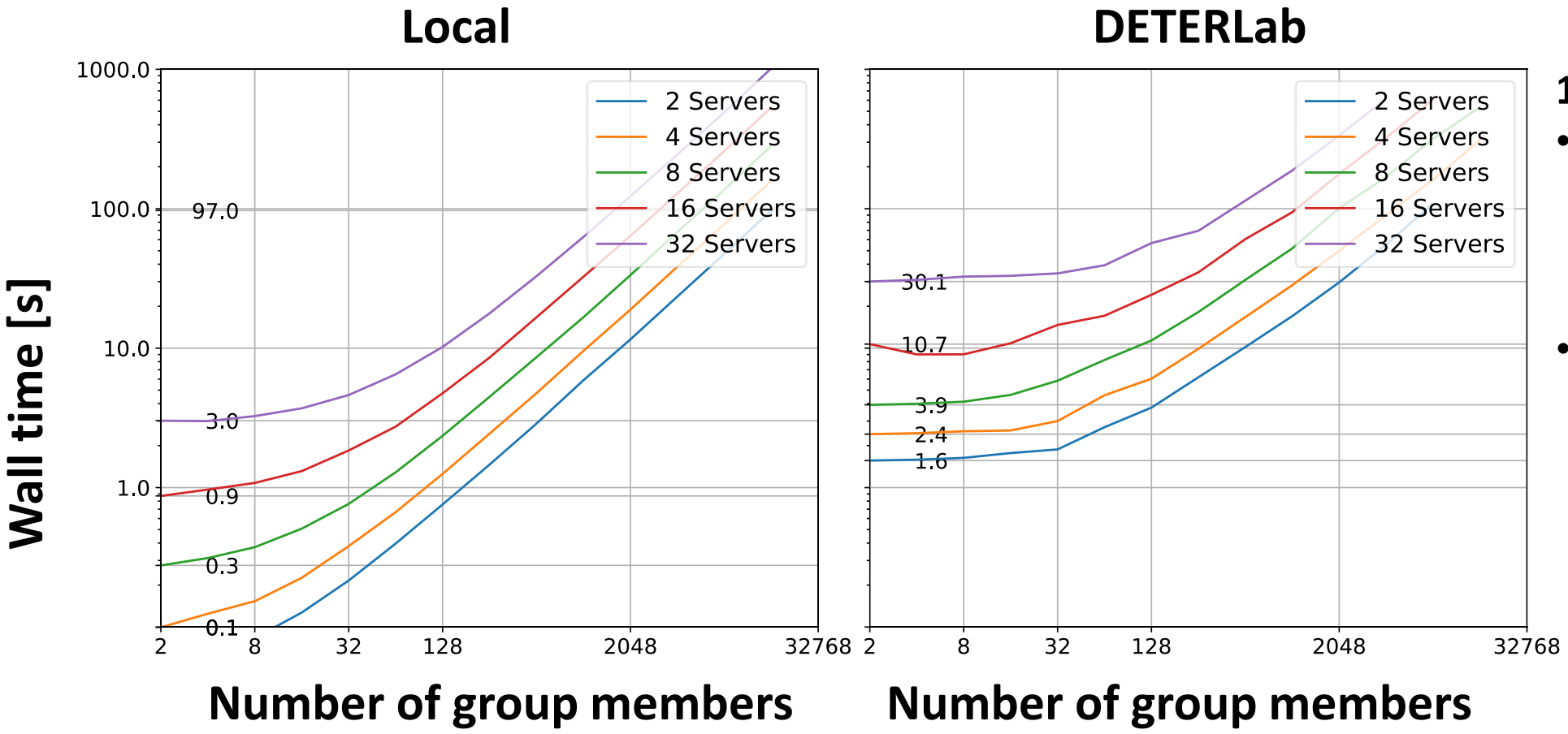


Need to keep state across endpoint calls
➔ avoid by storing it in clients

Cothority Implementation

- DAGA Library (continuation of A. Villard's work)
- New *Service & Protocols*
  (context generation / challenge generation / DAGA servers' protocol)
- Can run simulations locally and on DETERLab
- 80% code coverage
- Possible to generate proto files
- CLI client

Simulation results – total authentication time



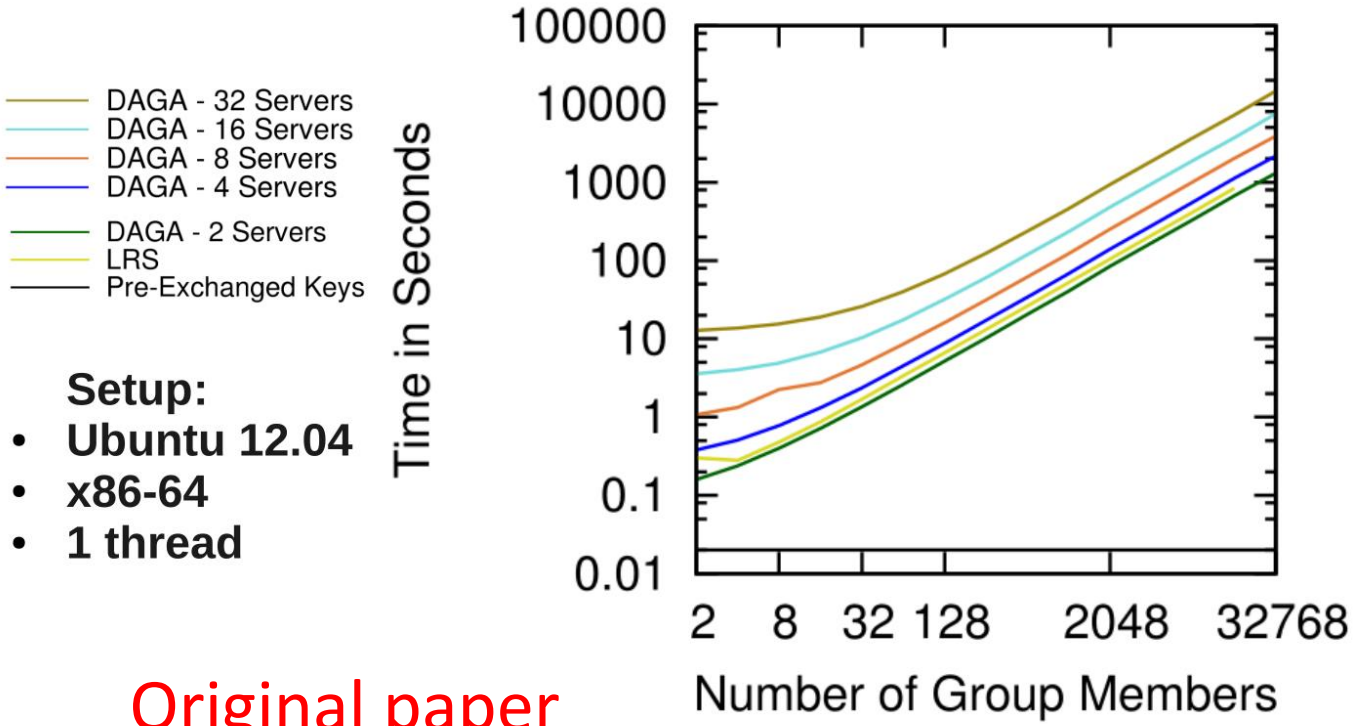
1) DETERLab Setup:

- ***pc2133*** **nodes:**
  - **Ubuntu 14.04, AMD64**
  - **CPU: 4 @ 2,13 GHz**
  - **RAM: 4 GiB**
- **LAN with 100 ms delay**

2) Local Setup:

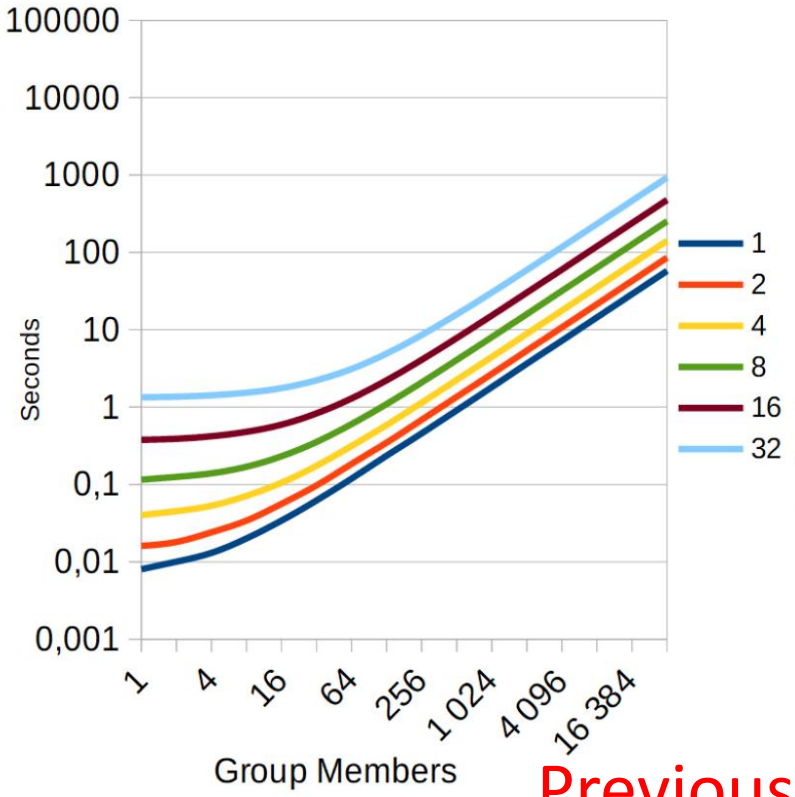
- **Debian 9, AMD64**
  CPU: 8 @ 2.50GHz
  RAM: 16 GiB

Original results and previous student's results



Setup:
- **Ubuntu 12.04**
- **x86-64**
- **1 thread**

Original paper (2014)

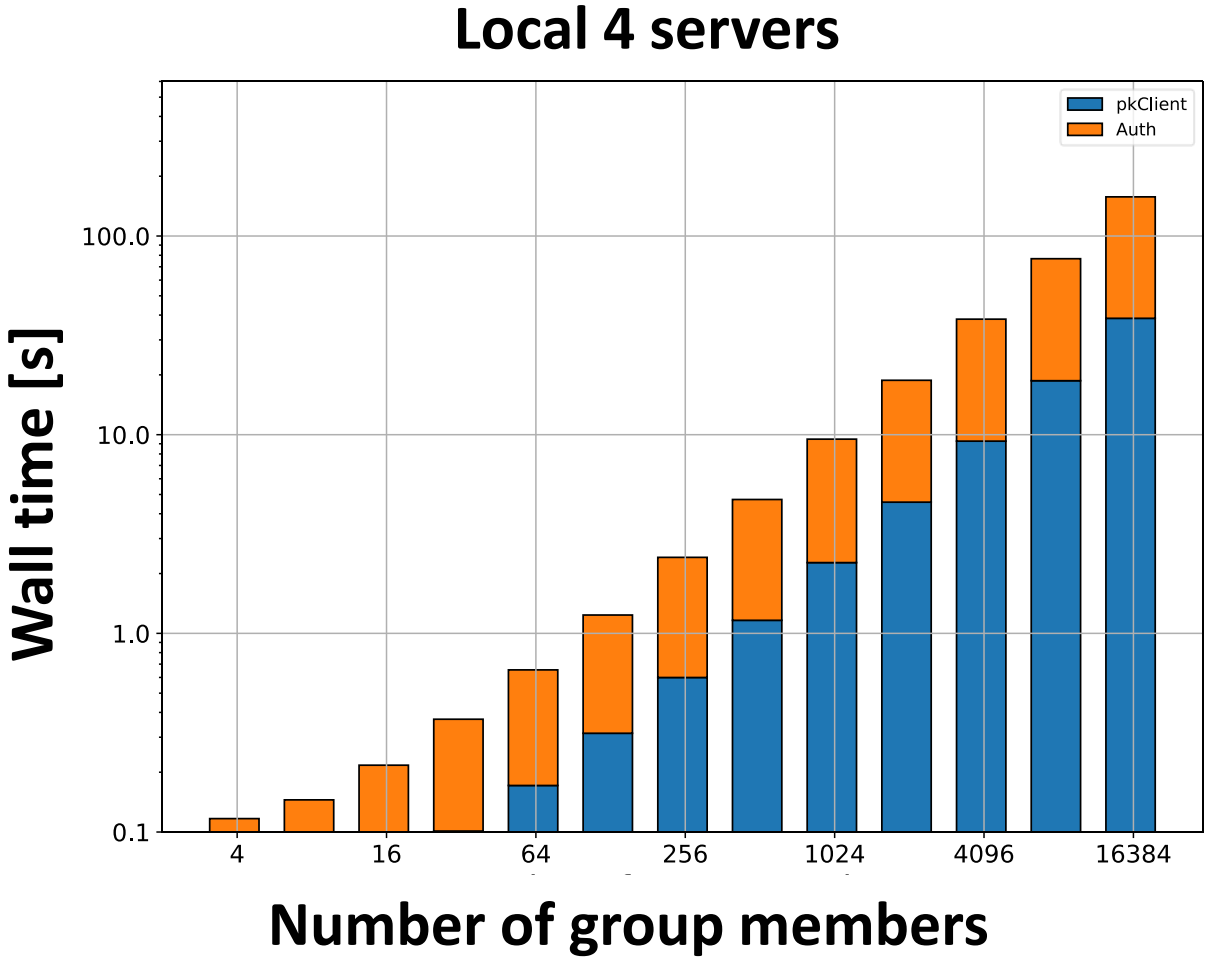


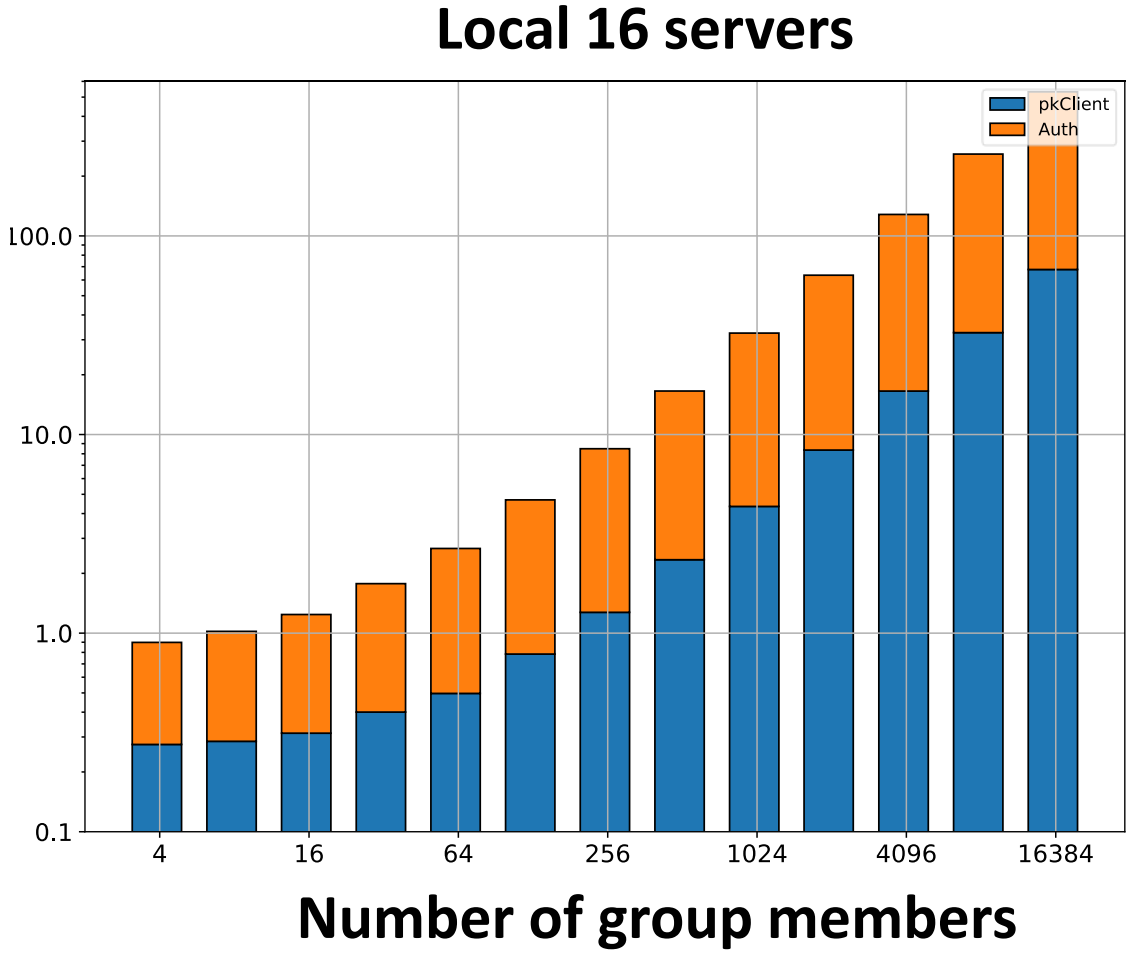
Setup:
- **Windows 10**
- **x86-64**
- **1 thread @4,5GHz**

Previous student

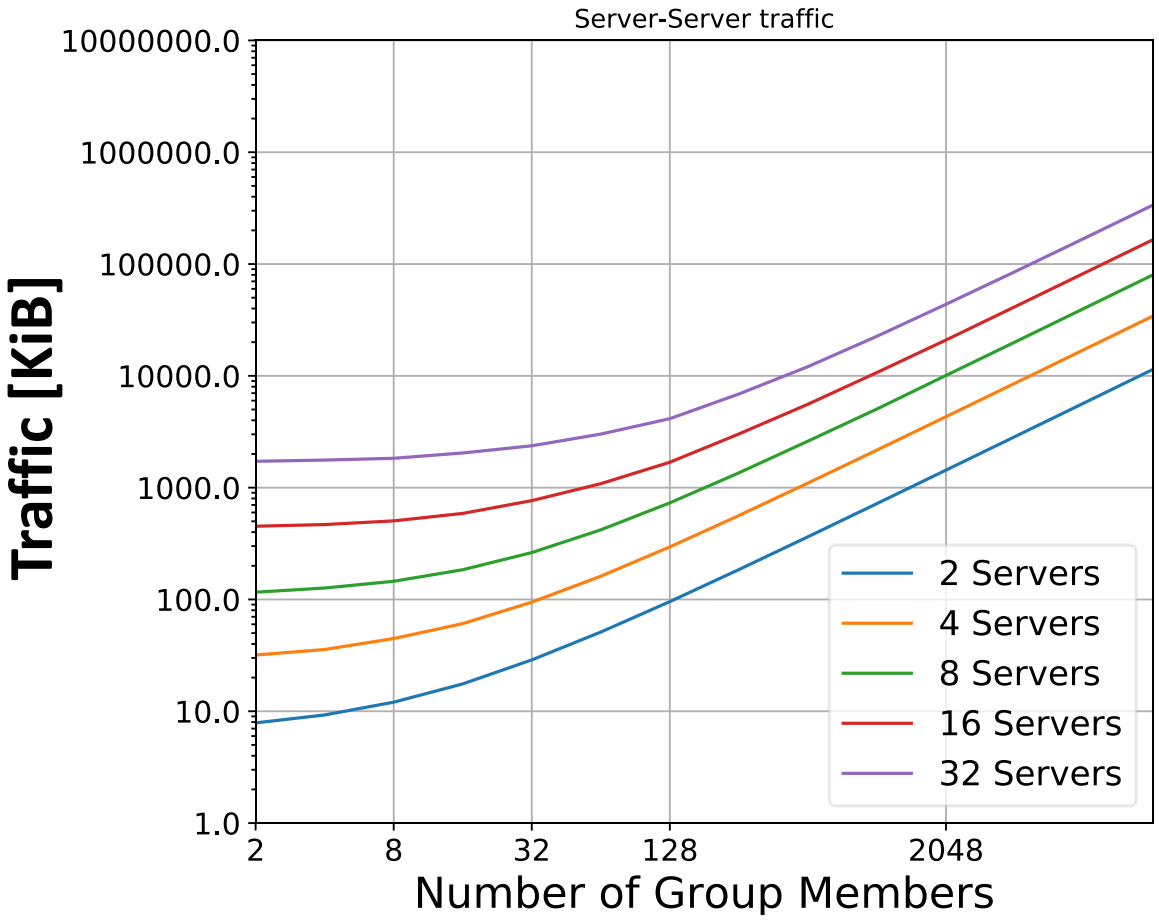
Taken from https://github.com/dedis/student_17/blob/master/pfs_pop/presentation_pfs_pop.pdf

Simulation results – total authentication time

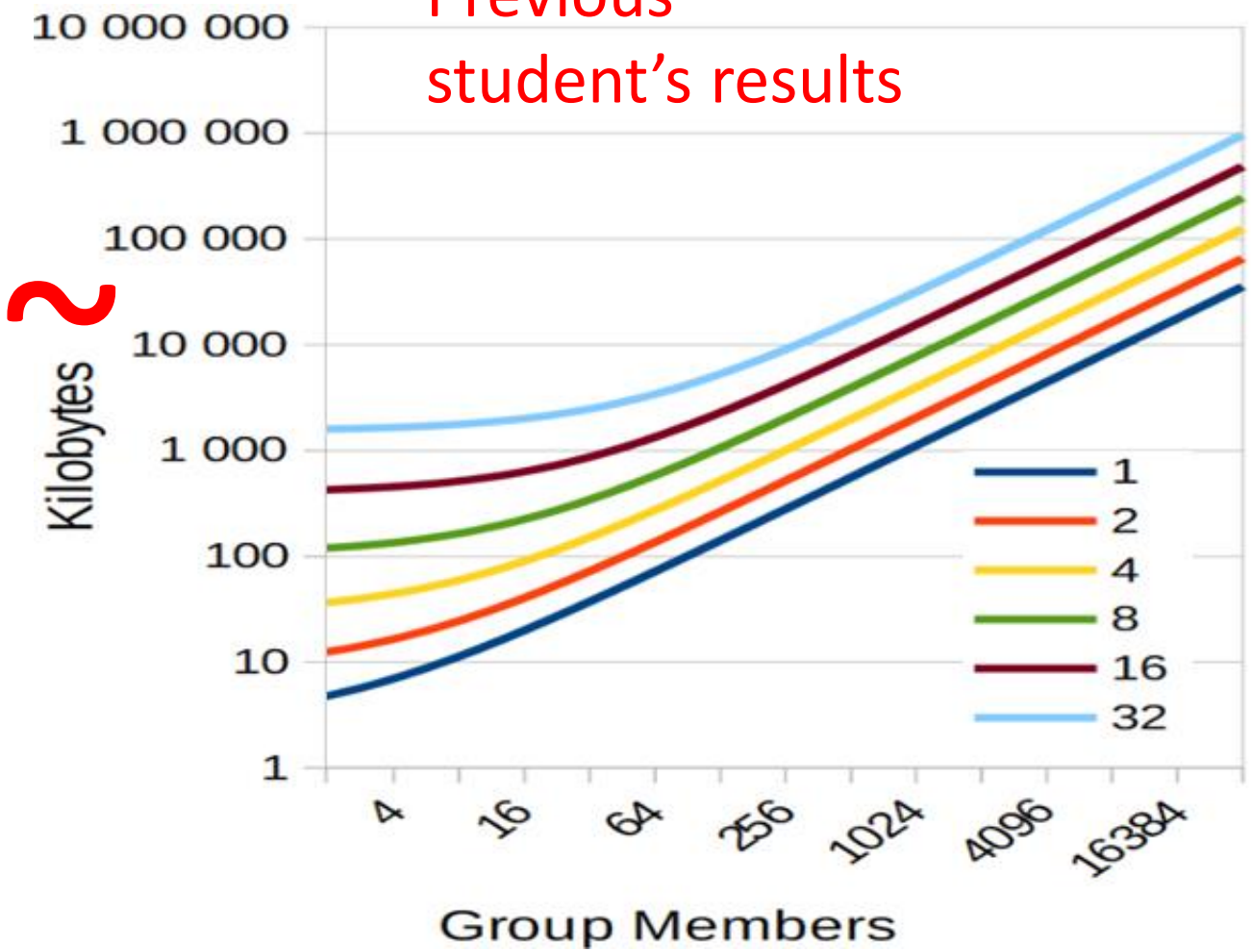




Simulation results – total server traffic



Previous student's results



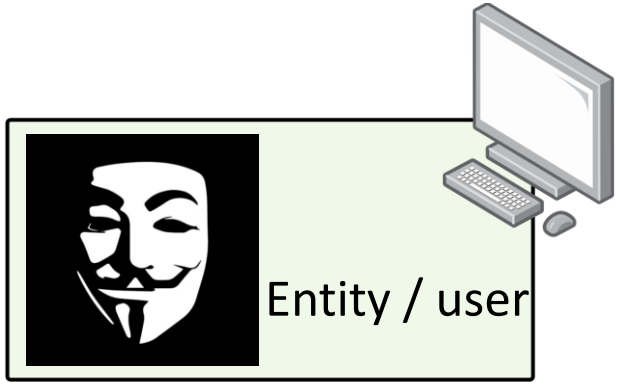
Cothority Implementation

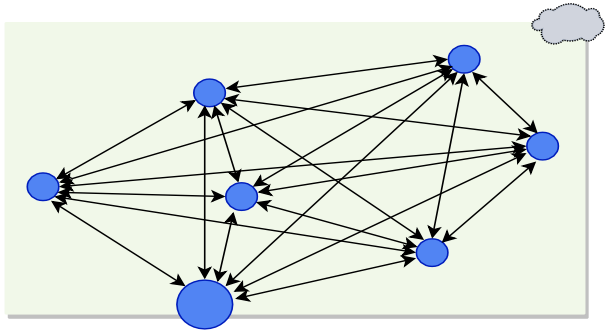
- DAGA Library (continuation of A. Villard's work)
- New *Service & Protocols*
  (context generation / challenge generation / DAGA servers' protocol)
- Can run simulations locally and on DETERLab
- 80% code coverage
- Possible to generate proto files
- CLI client

Overview

- Background / DAGA
- Cothority implementation
- Authentication delegation
- PoC demo
- Conclusion &? Future

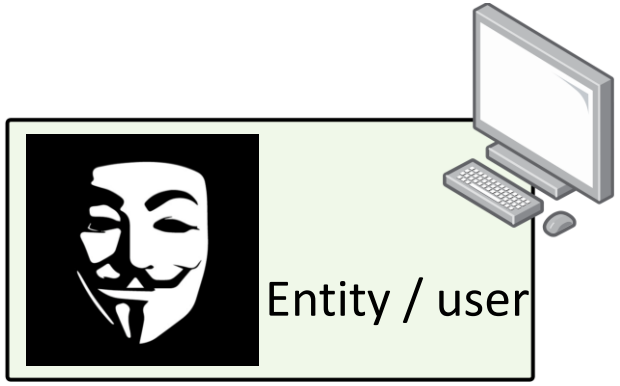
Authentication delegation

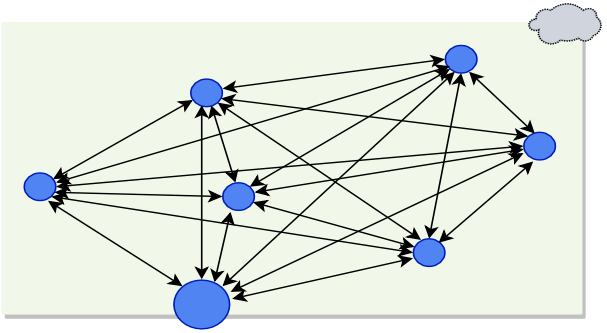




DAGA cothority

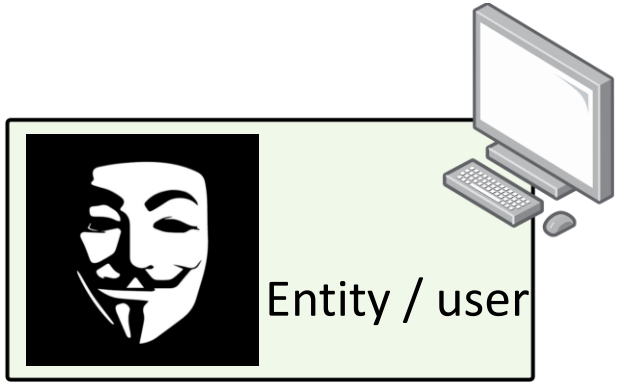
Authentication delegation

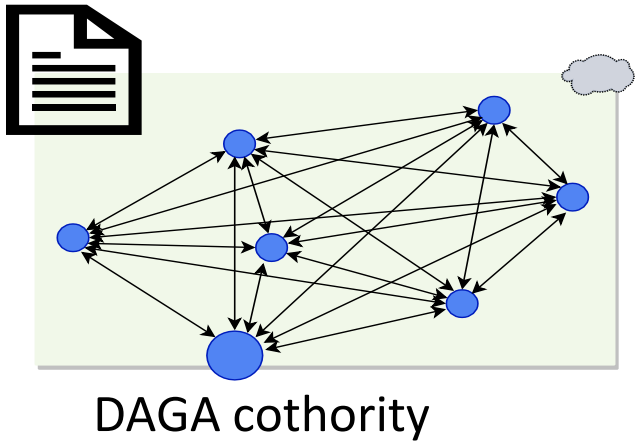




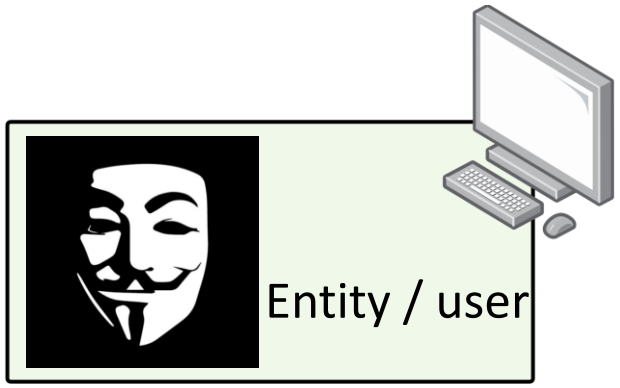
DAGA cothority

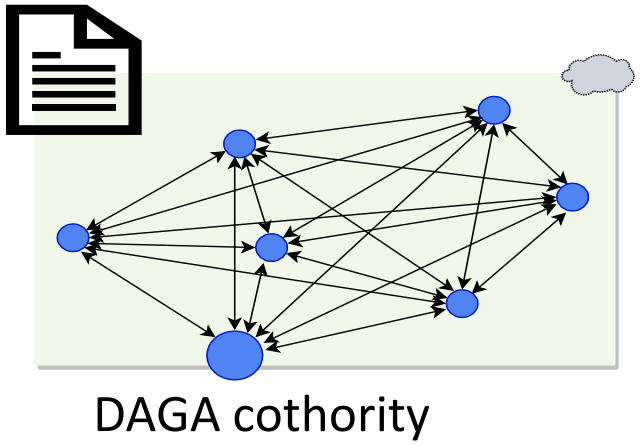
Authentication delegation

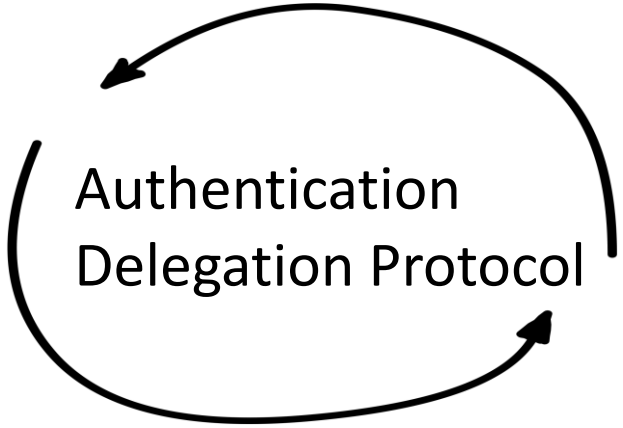


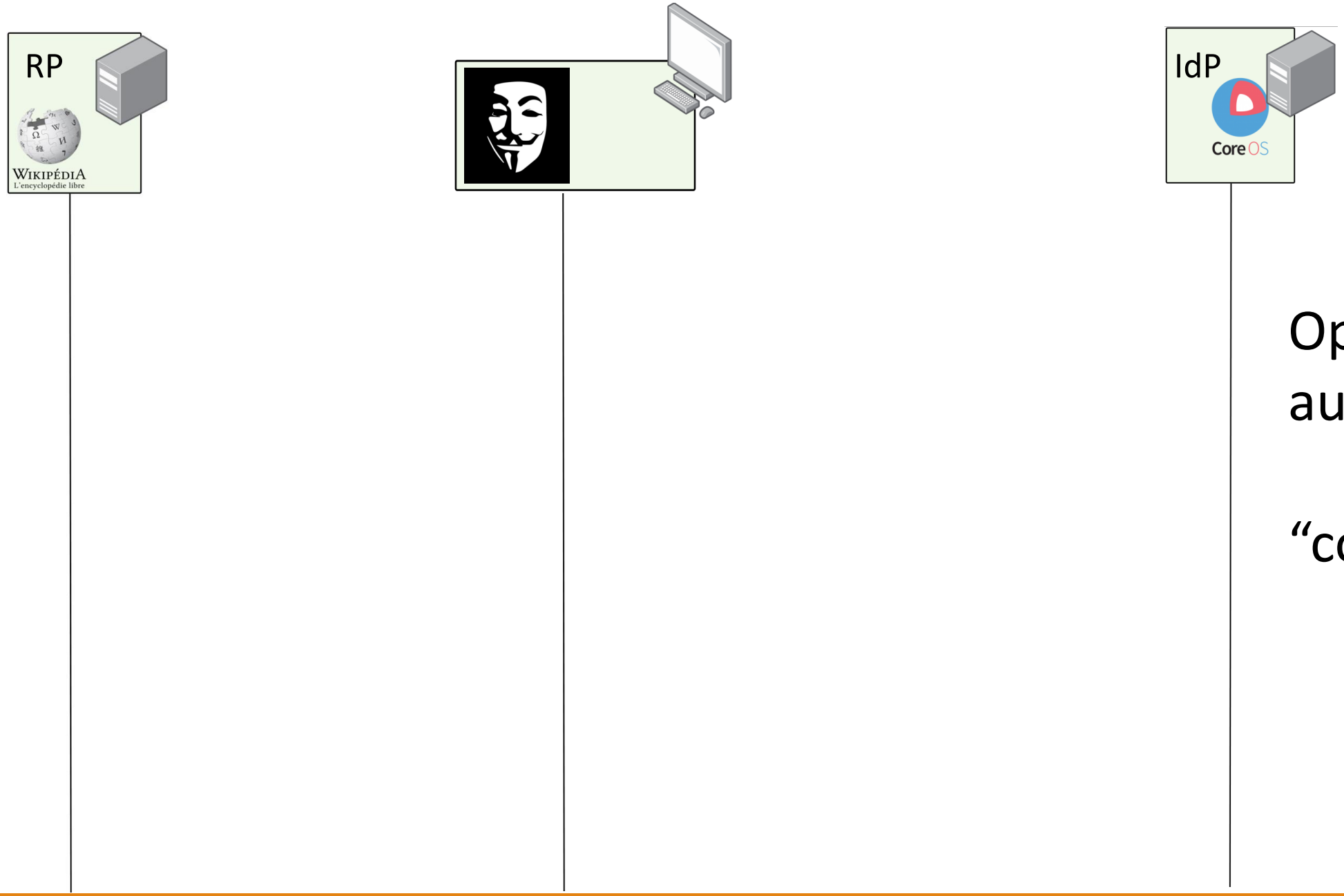


Authentication delegation





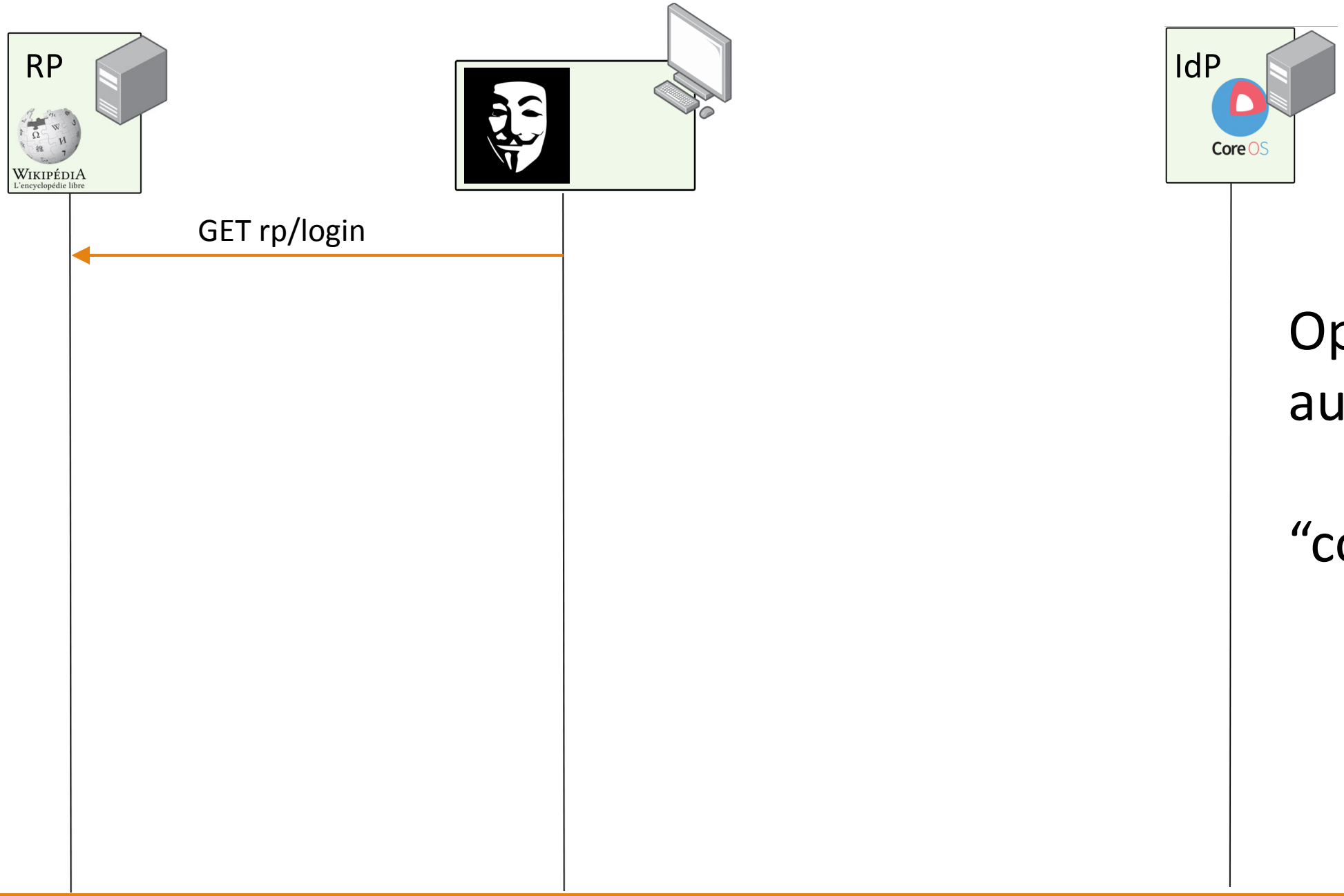




OpenID connect authentication

-

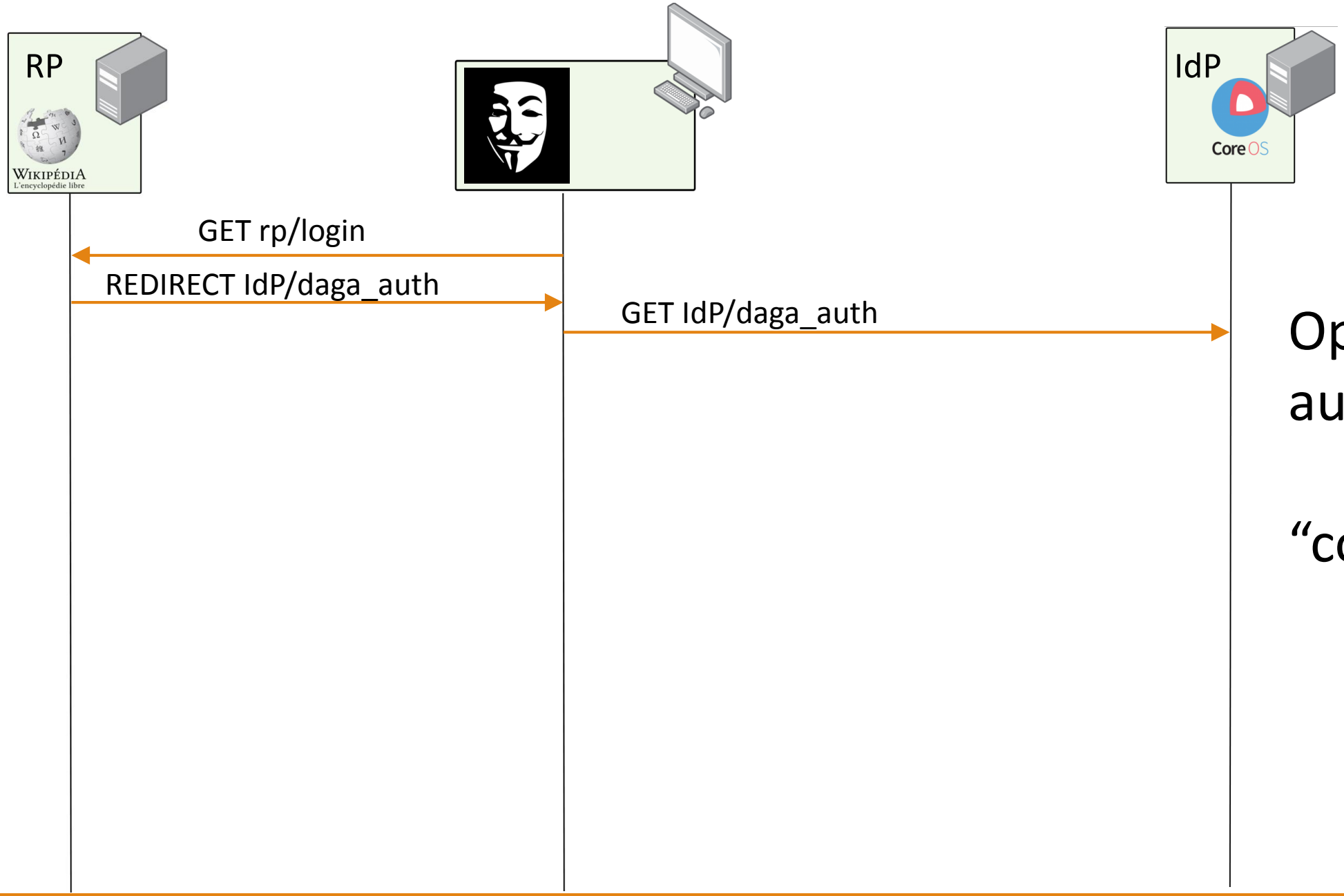
“code flow”



OpenID connect authentication

-

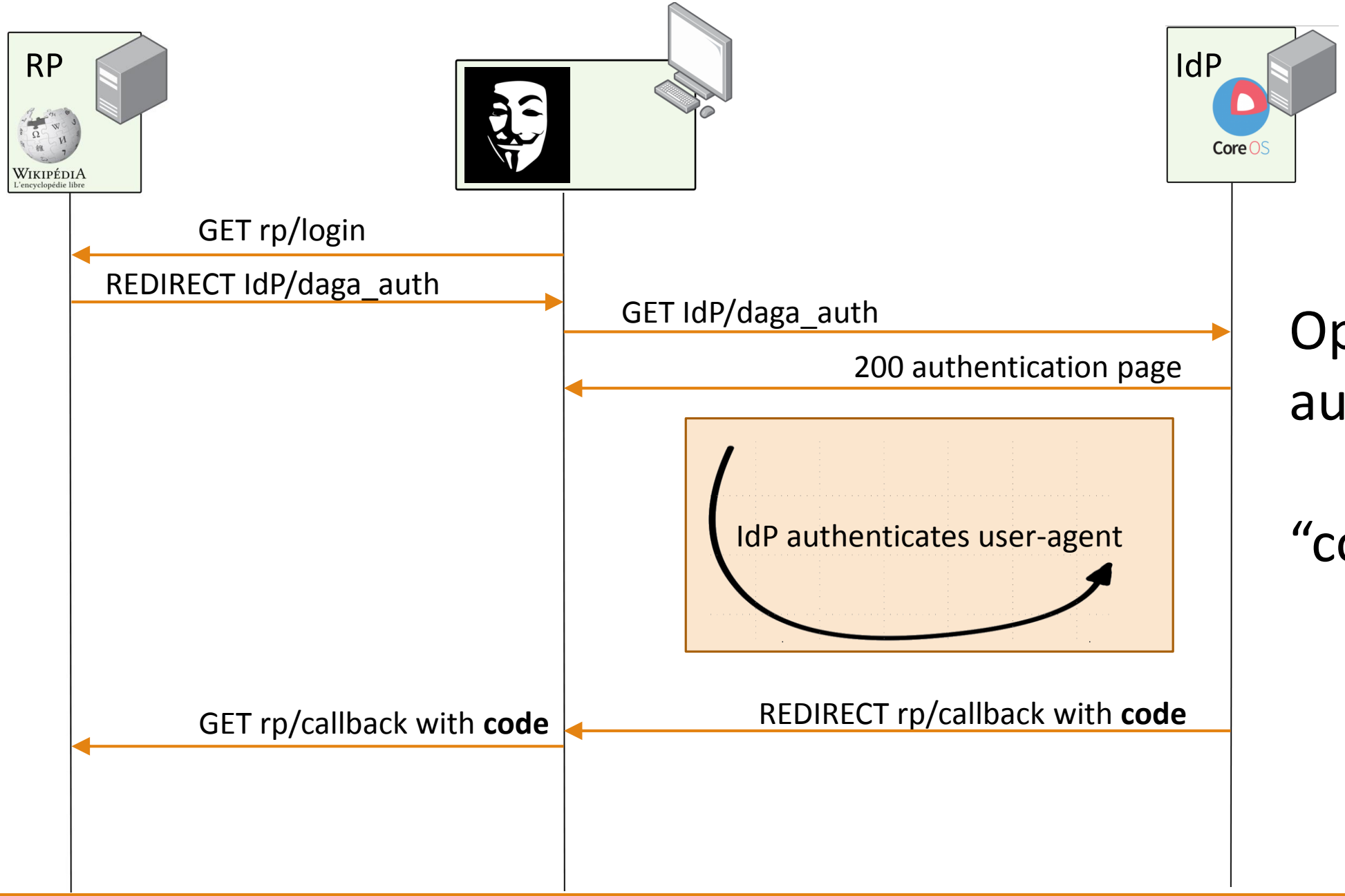
“code flow”



OpenID connect authentication

-

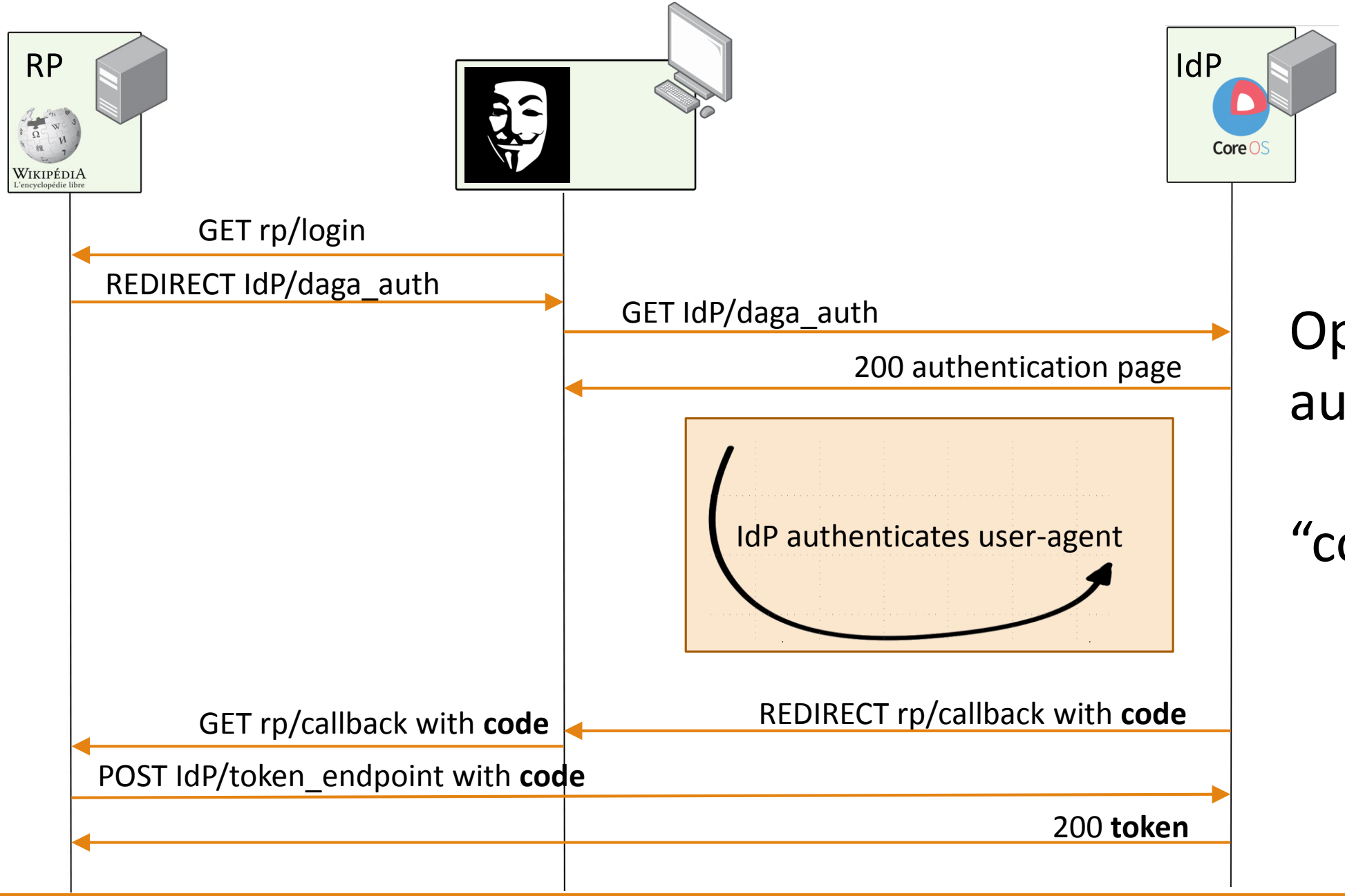
“code flow”



OpenID connect authentication

-

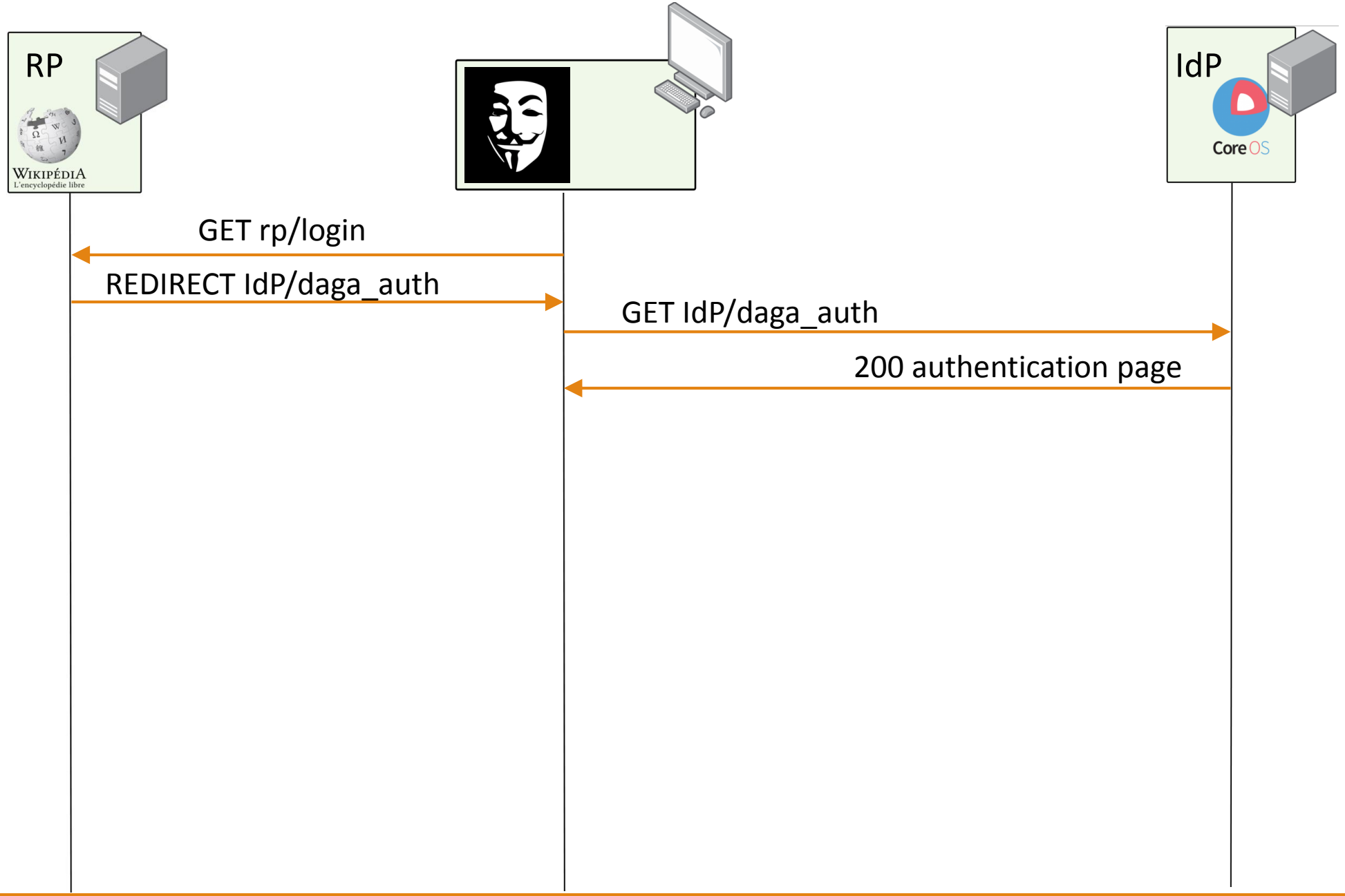
“code flow”

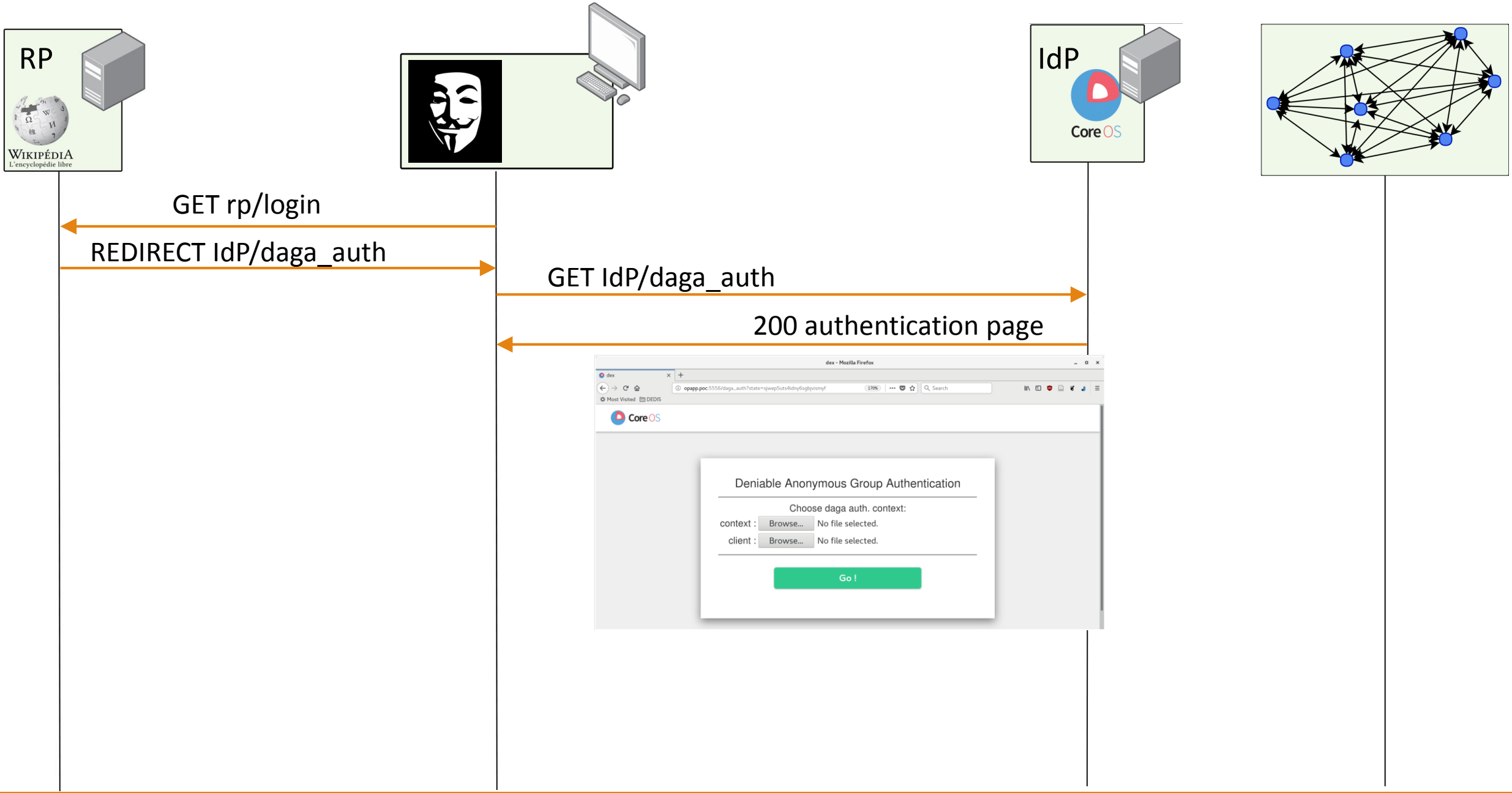


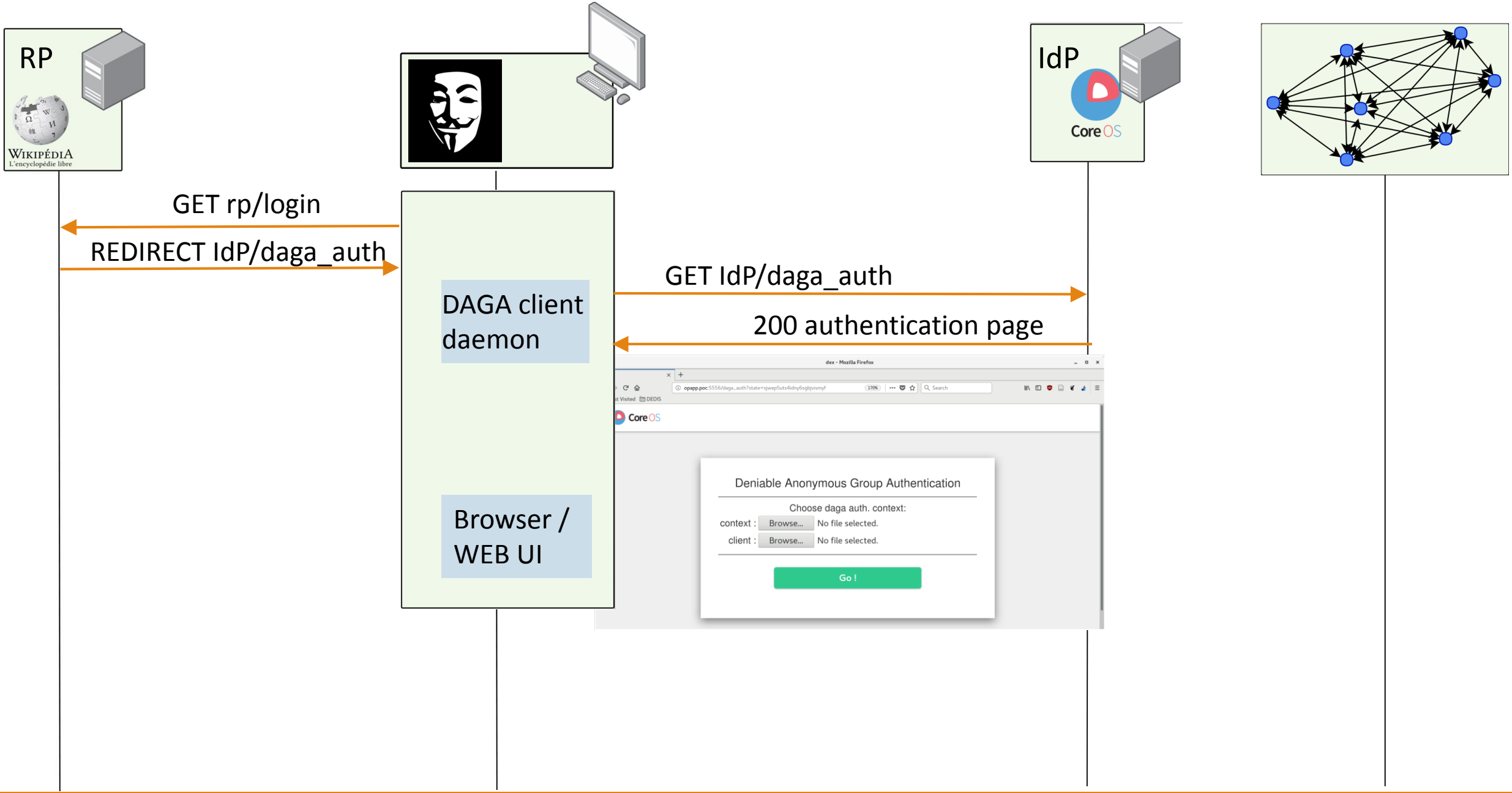
OpenID connect authentication

-

“code flow”


RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
GET rp/login
REDIRECT IdP/daga_auth
GET IdP/daga_auth
200 authentication page


RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
GET rp/login
REDIRECT IdP/daga_auth
GET IdP/daga_auth
200 authentication page
dex - Mozilla Firefox
CoreOS
Deniable Anonymous Group Authentication
Choose daga auth. context:
context : Browse... No file selected.
client : Browse... No file selected.
Go !


RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
GET rp/login
REDIRECT IdP/daga_auth
DAGA client daemon
GET IdP/daga_auth
200 authentication page
Browser / WEB UI
Deniable Anonymous Group Authentication
Choose daga auth. context:
context :
Browse...
No file selected.
client :
Browse...
No file selected.
Go !

RP

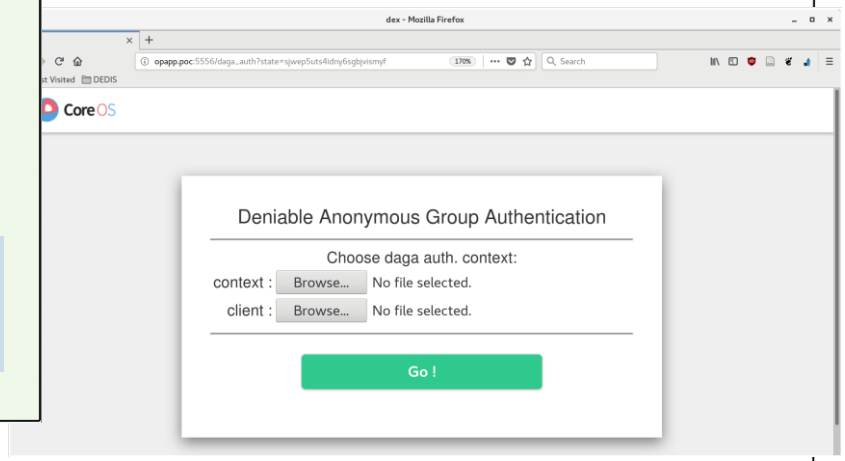
WIKIPÉDIA
L'encyclopédie libre

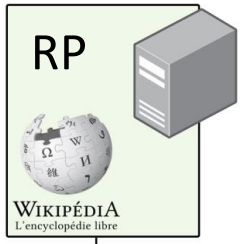
DAGA client daemon

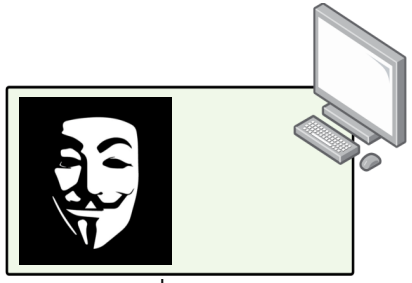
Browser / WEB UI

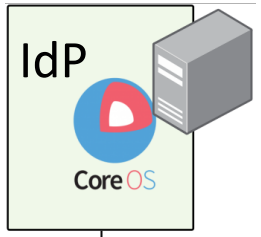
IdP

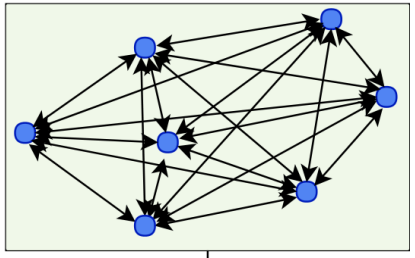
CoreOS







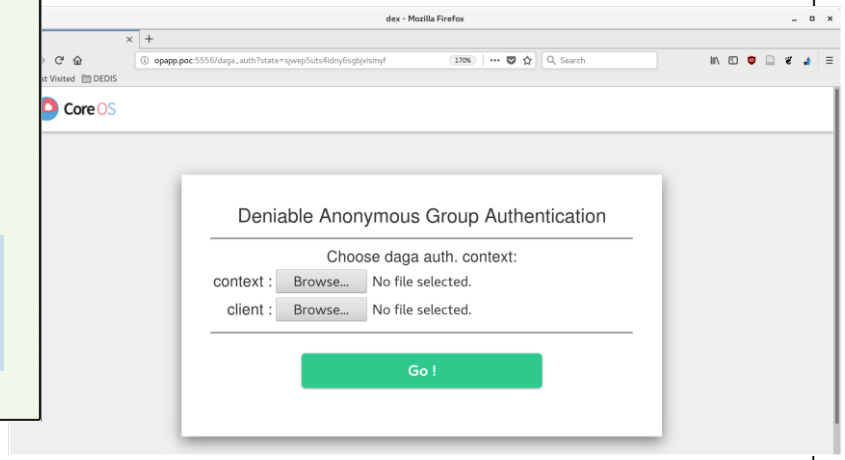


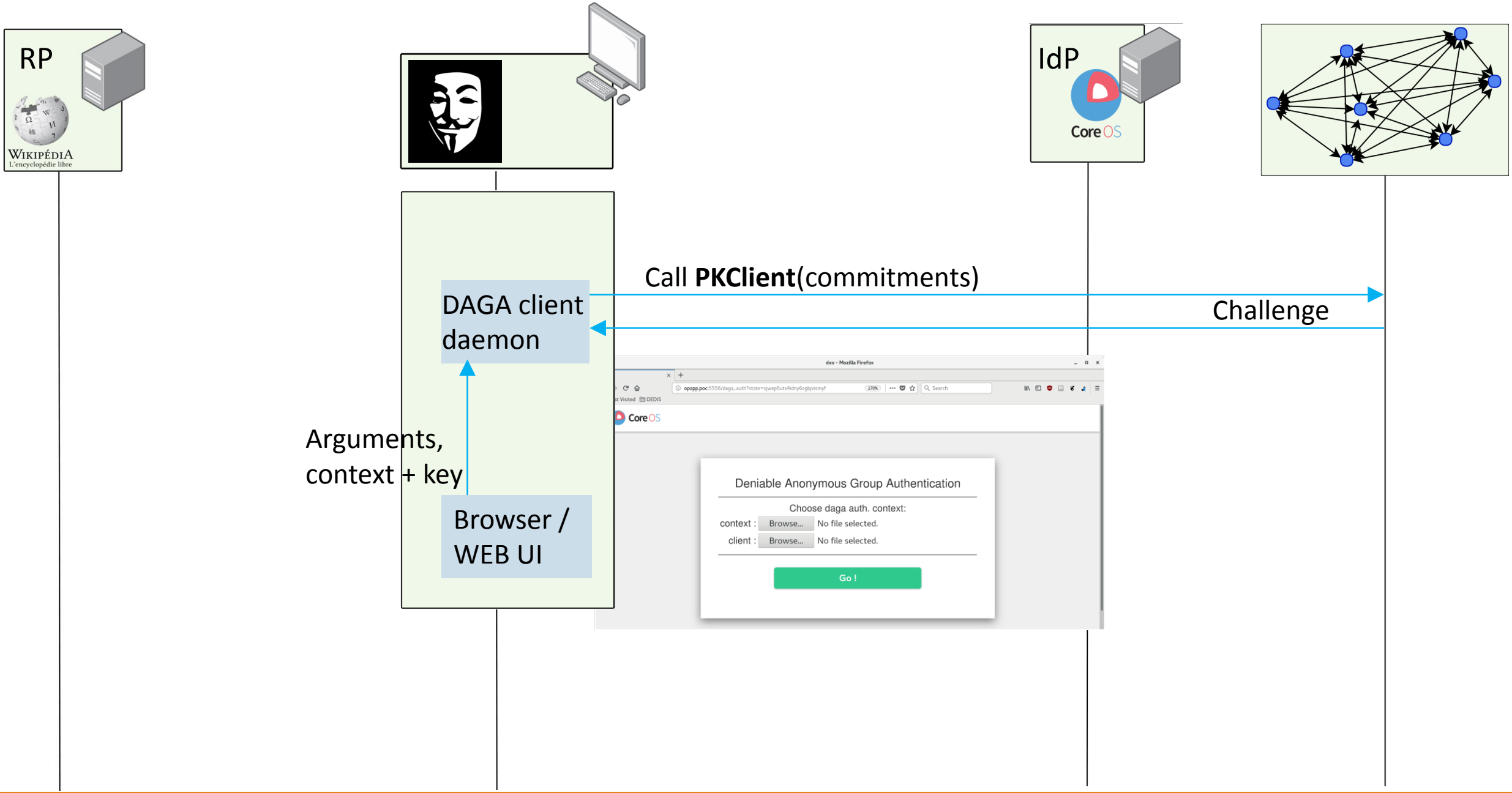


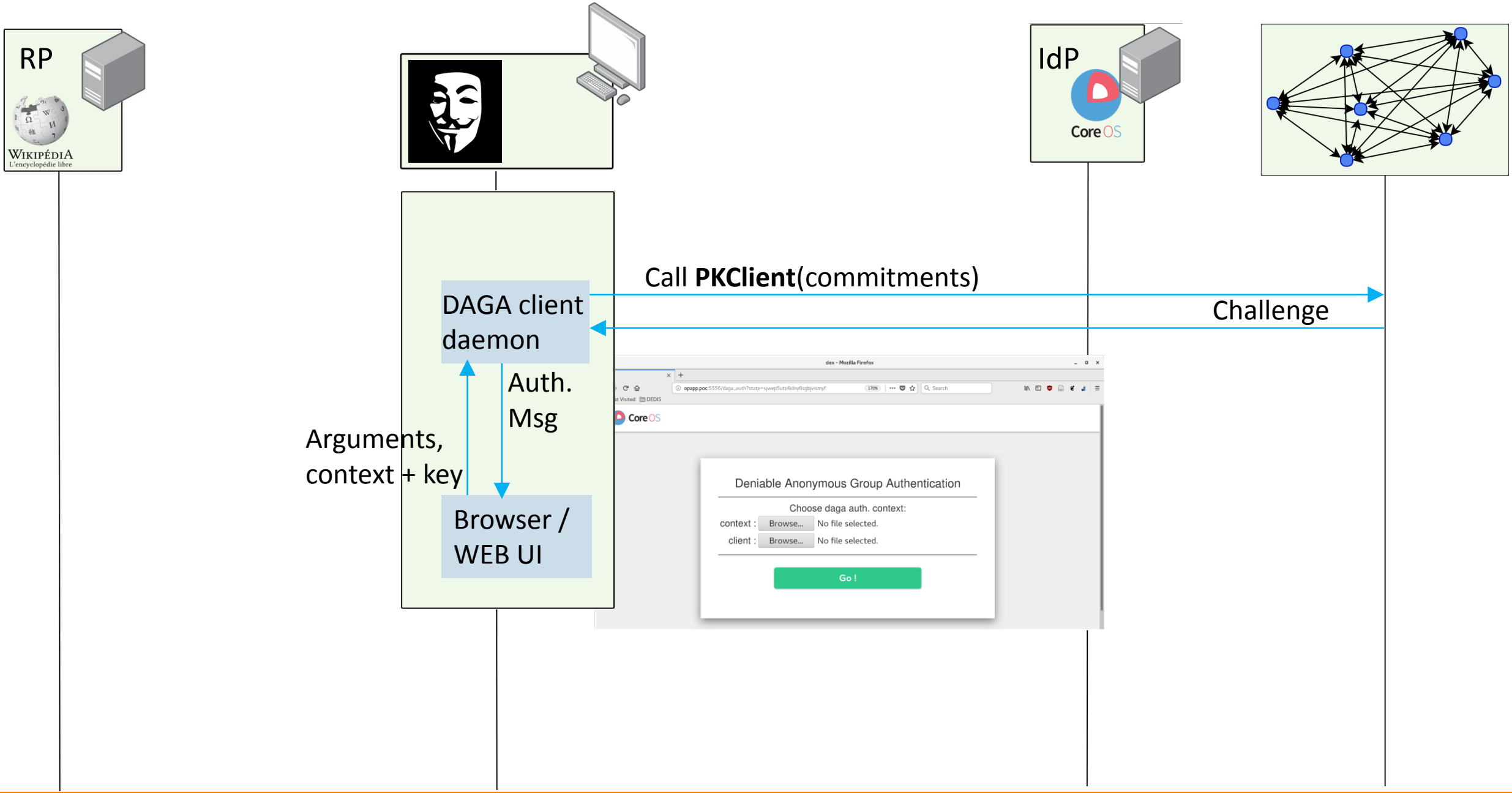
DAGA client daemon

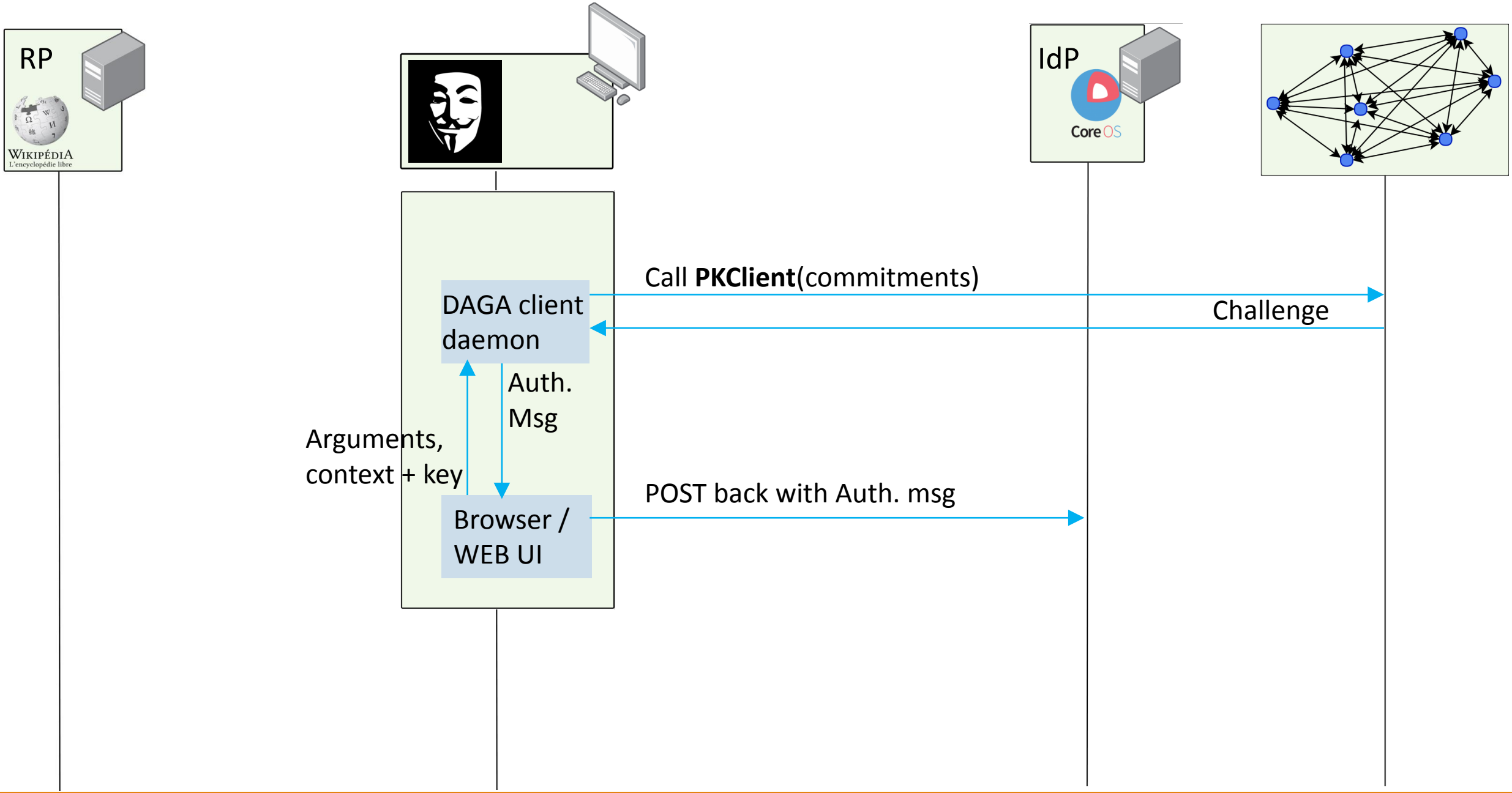
Arguments, context + key

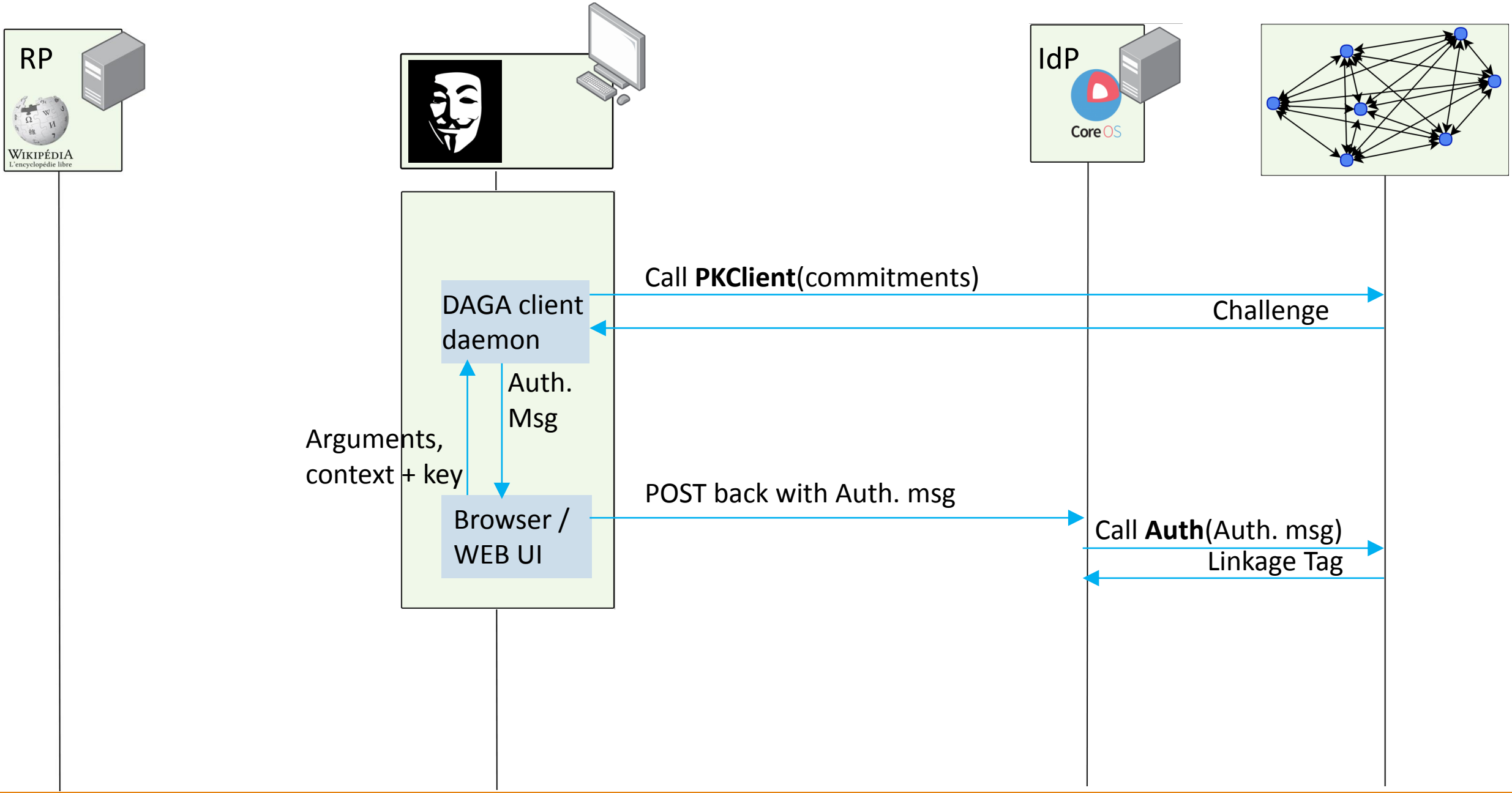
Browser / WEB UI

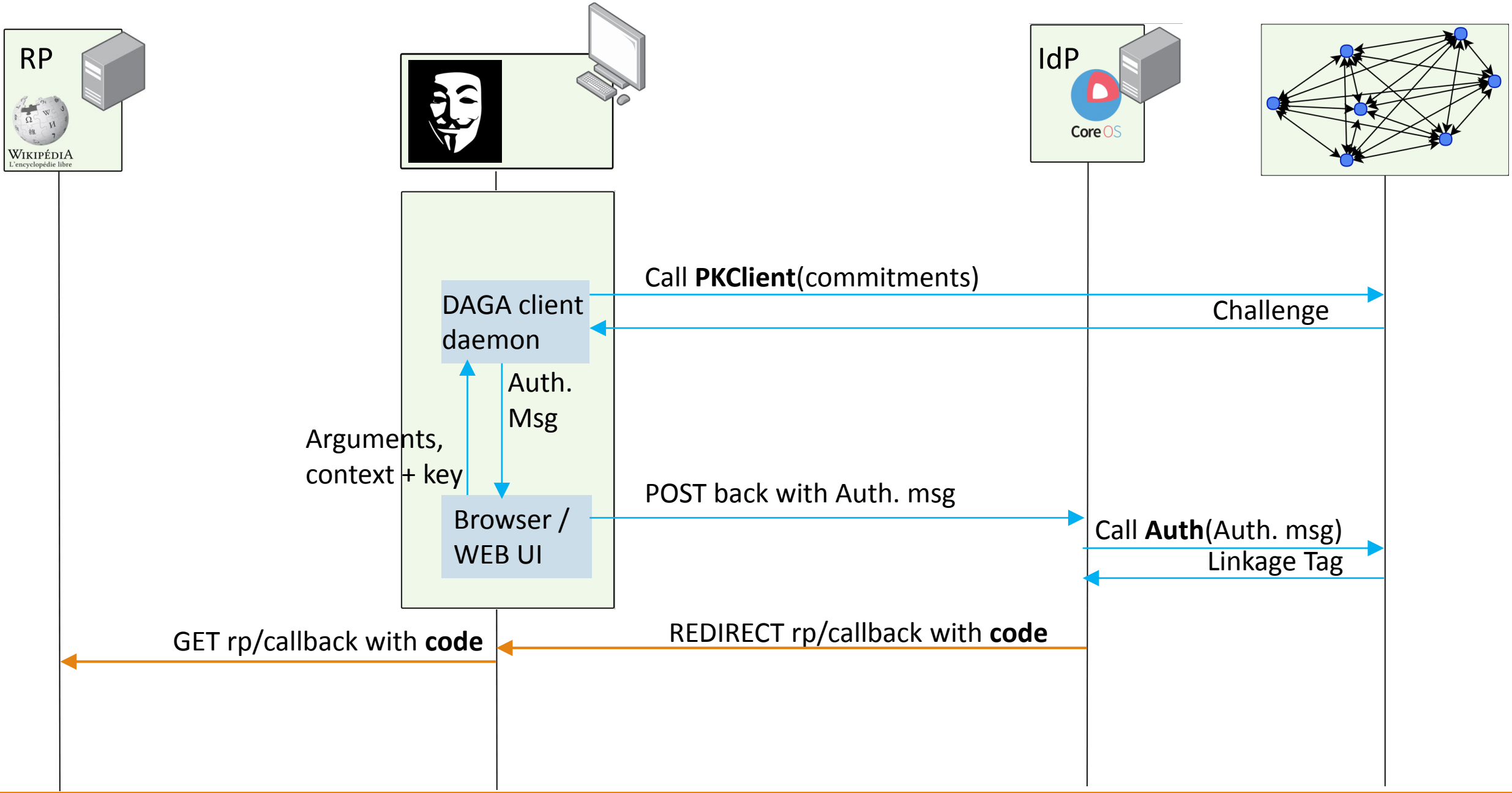


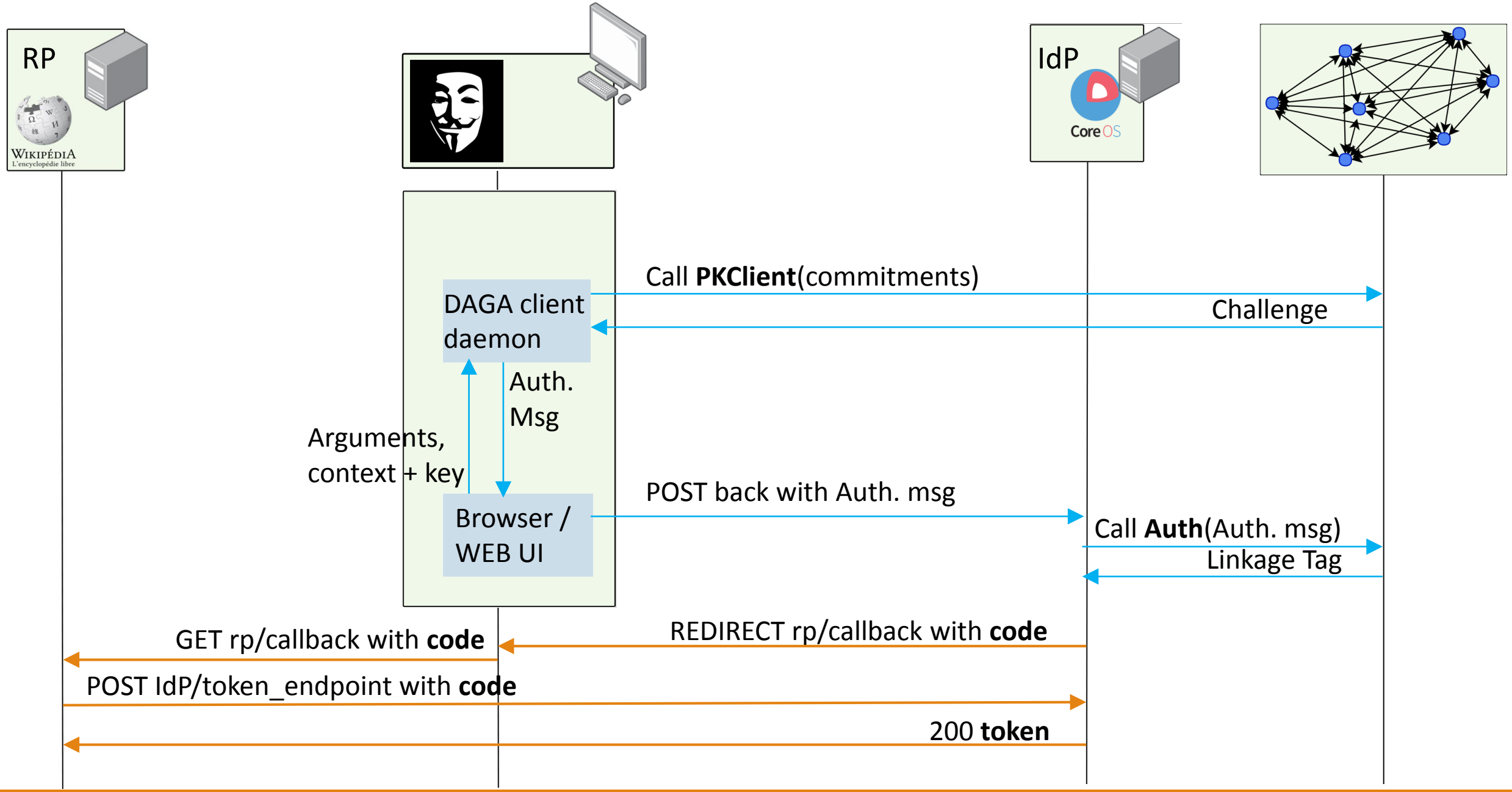

RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
Call **PKClient**(commitments)
Challenge
DAGA client daemon
Arguments, context + key
Browser / WEB UI
dex - Mozilla Firefox
CoreOS
Deniable Anonymous Group Authentication
Choose daga auth. context:
context : Browse... No file selected.
client : Browse... No file selected.
Go !


RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
Call **PKClient**(commitments)
Challenge
DAGA client daemon
Auth. Msg
Arguments, context + key
Browser / WEB UI
Deniable Anonymous Group Authentication
Choose daga auth. context:
context : Browse... No file selected.
client : Browse... No file selected.
Go !


RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
DAGA client daemon
Call **PKClient**(commitments)
Challenge
Auth. Msg
Arguments, context + key
POST back with Auth. msg
Browser / WEB UI


RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
DAGA client daemon
Auth. Msg
Arguments, context + key
Browser / WEB UI
Call **PKClient**(commitments)
Challenge
POST back with Auth. msg
Call **Auth**(Auth. msg)
Linkage Tag


RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
DAGA client daemon
Auth. Msg
Arguments, context + key
Browser / WEB UI
Call **PKClient**(commitments)
Challenge
POST back with Auth. msg
Call **Auth**(Auth. msg)
Linkage Tag
REDIRECT rp/callback with **code**
GET rp/callback with **code**


RP
WIKIPÉDIA
L'encyclopédie libre
IdP
CoreOS
DAGA client daemon
Auth. Msg
Arguments, context + key
Browser / WEB UI
Call **PKClient**(commitments)
Challenge
POST back with Auth. msg
Call **Auth**(Auth. msg)
Linkage Tag
REDIRECT rp/callback with **code**
GET rp/callback with **code**
POST IdP/token_endpoint with **code**
200 **token**

Demo

Conclusion

- Democratization of DAGA as anonymous authentication is feasible
- Future works:

Conclusion

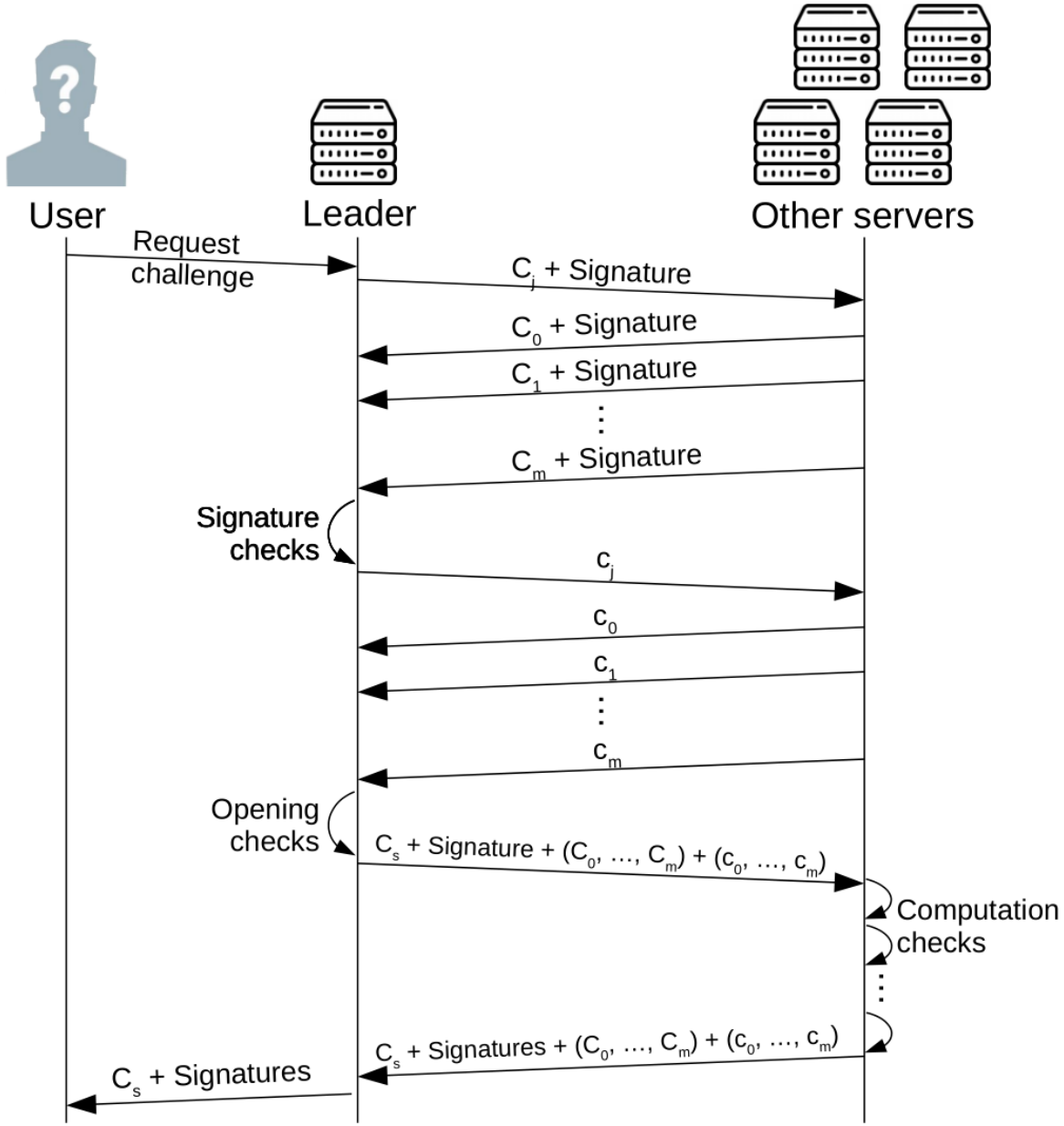
- Democratization of DAGA as anonymous authentication is feasible
- Future works:
  - Need ways to manage partnerships and evolve contexts

Conclusion

- Democratization of DAGA as anonymous authentication is feasible
- Future works:
  - Need ways to manage partnerships and evolve contexts
  - Need ways to scale (random sub-groups)

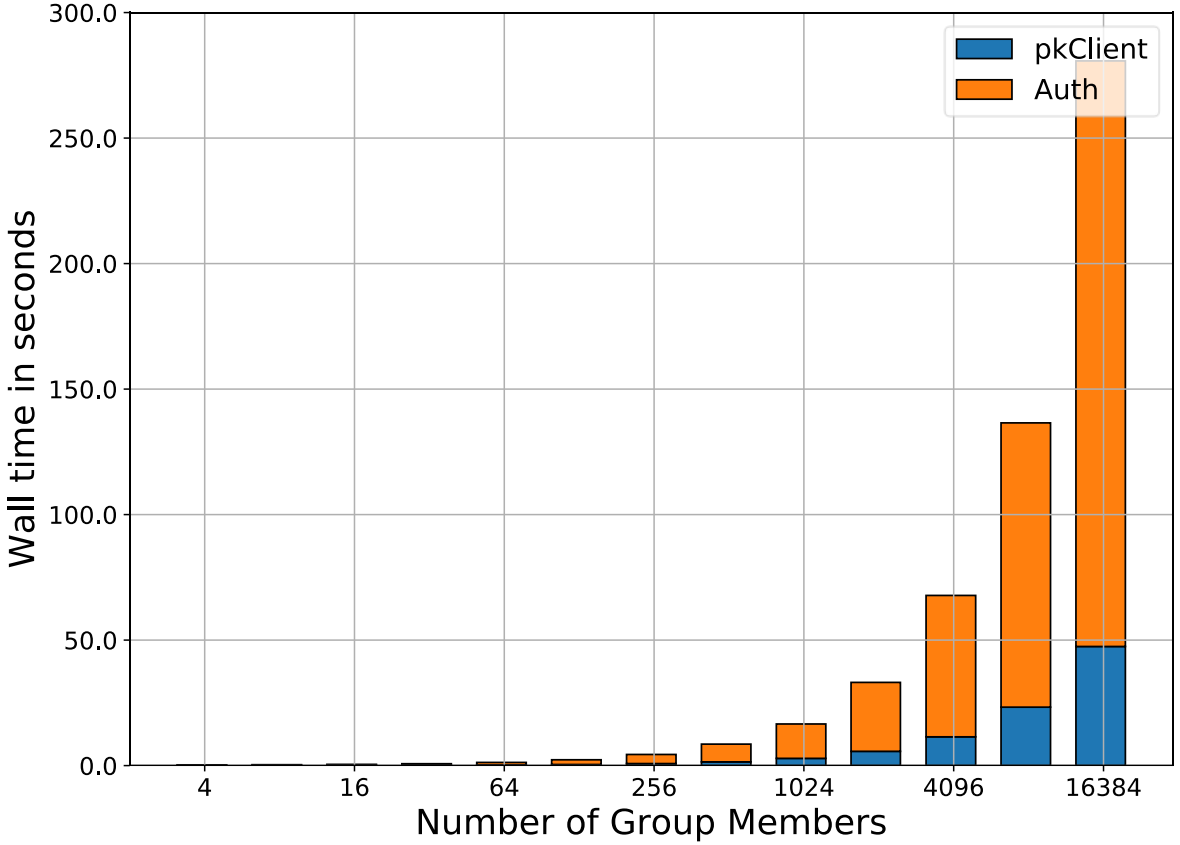
Conclusion

- Democratization of DAGA as anonymous authentication is feasible
- Future works:
  - Need ways to manage partnerships and evolve contexts
  - Need ways to scale (random sub-groups)
  - Need to armor everything (memory protection,…)

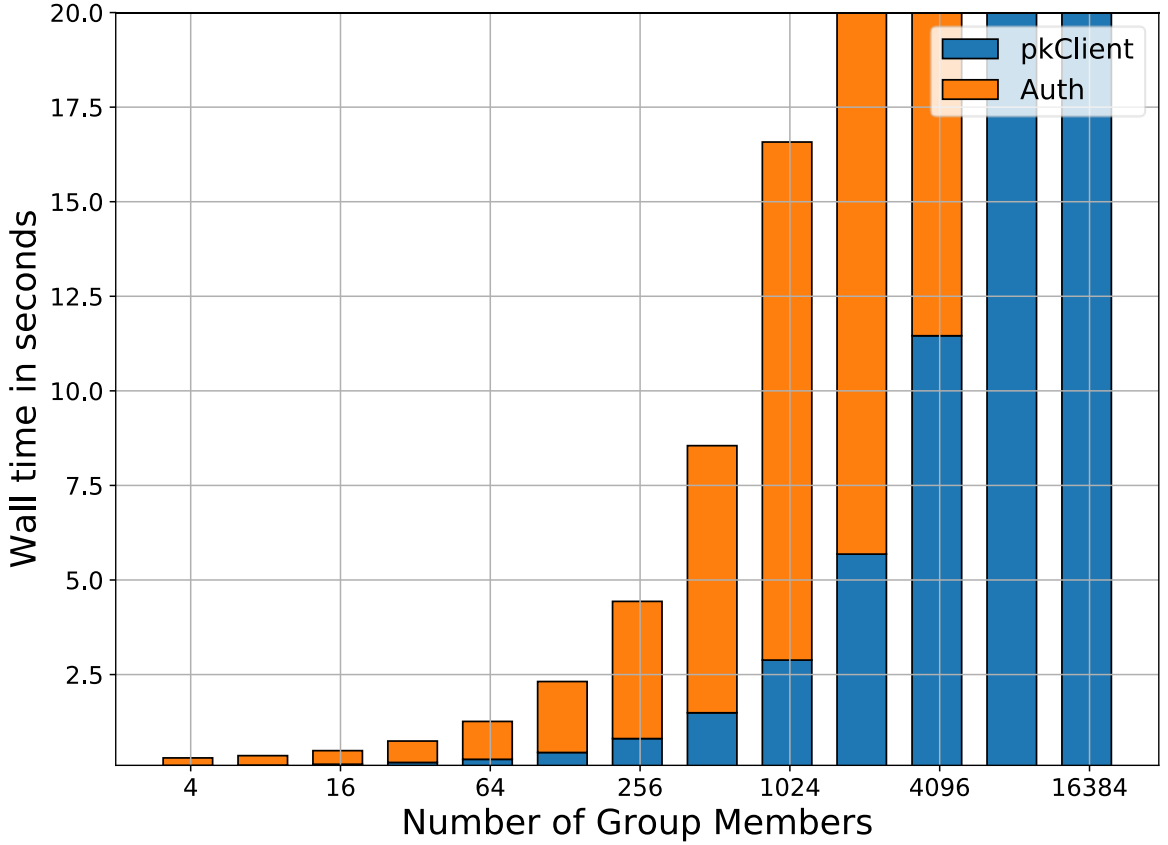


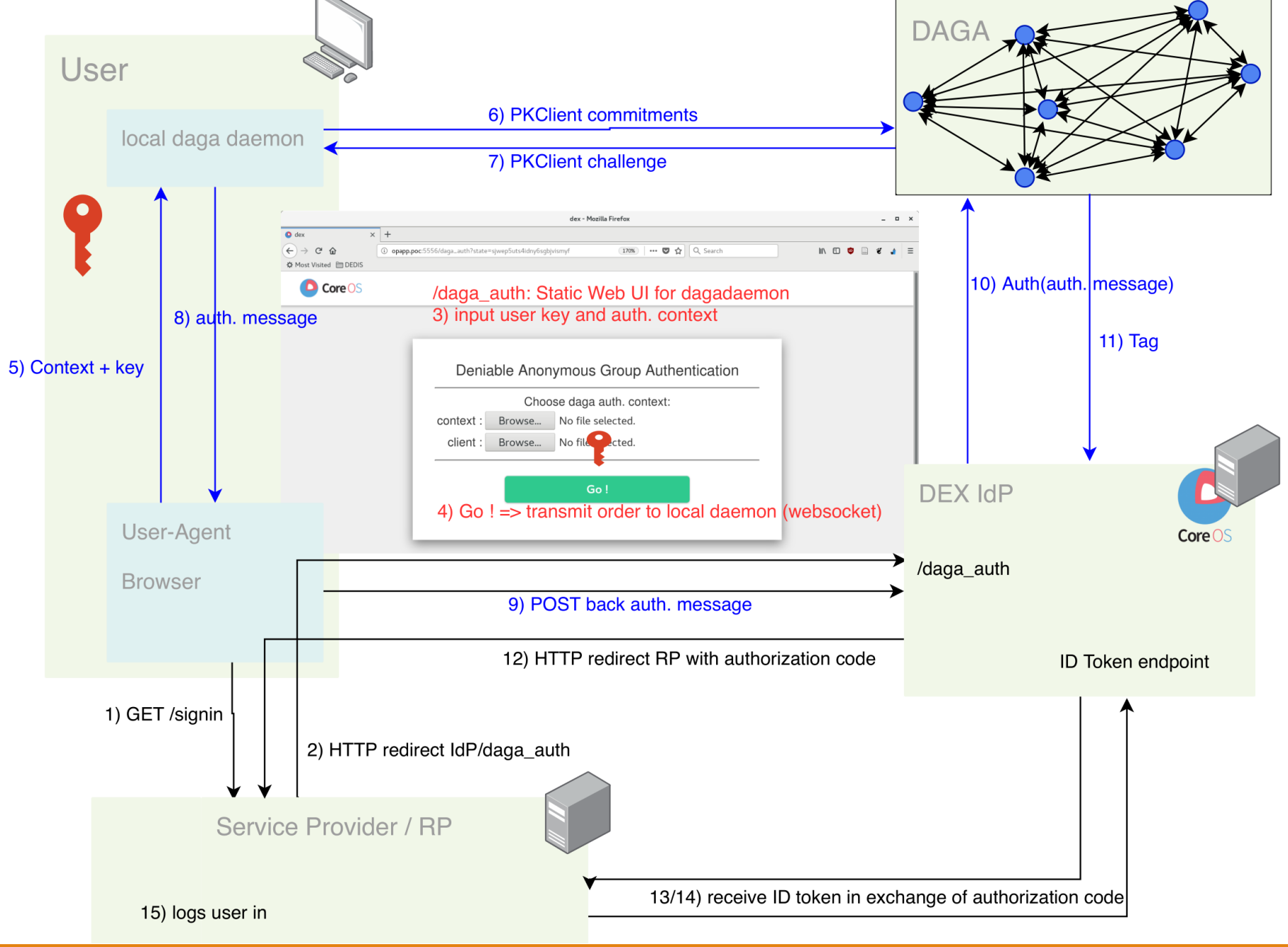
Taken from https://github.com/dedis/student_17/blob/master/pfs_pop/report_pfs_pop.pdf

Local 8 servers, linear



Local 8 servers, linear




User
local daga daemon
DAGA
6) PKClient commitments
7) PKClient challenge
5) Context + key
8) auth. message
/daga_auth: Static Web UI for dagadaemon
3) input user key and auth. context
Deniable Anonymous Group Authentication
Choose daga auth. context:
context :
Browse...
No file selected.
client :
Browse...
Go !
4) Go ! => transmit order to local daemon (websocket)
10) Auth(auth. message)
11) Tag
User-Agent
Browser
DEX IdP
/daga_auth
9) POST back auth. message
12) HTTP redirect RP with authorization code
ID Token endpoint
1) GET /signin
2) HTTP redirect IdP/daga_auth
Service Provider / RP
13/14) receive ID token in exchange of authorization code
15) logs user in