# Tutamen: A Next-Generation Secret-Storage System

Andy Sayler, Taylor Andrews, Matt Monaco, and Dirk Grunwald

Presented by Andy Sayler

SoCC 2016
10/06/16

University of Colorado **Boulder**

SFg5asknmc6e

SFg5asknmc6e

SFg5asknmc6e

Please Login

*Use your CS Moodle (i.e. IdentiKey) Credentials*

DTrump

GreatPassword

Login

# Secrets

SFg5asknmc6e

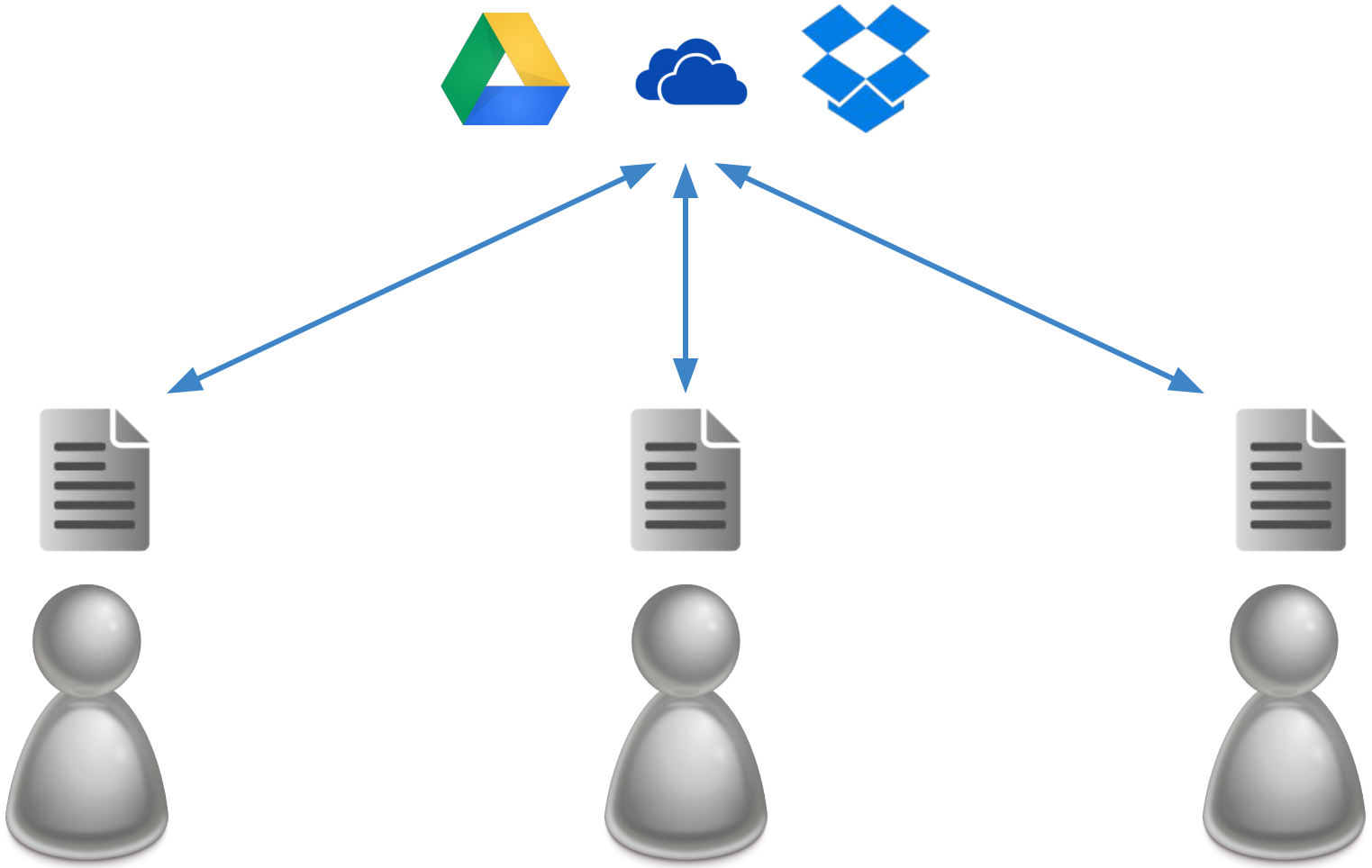
DTrump

GreatPassword

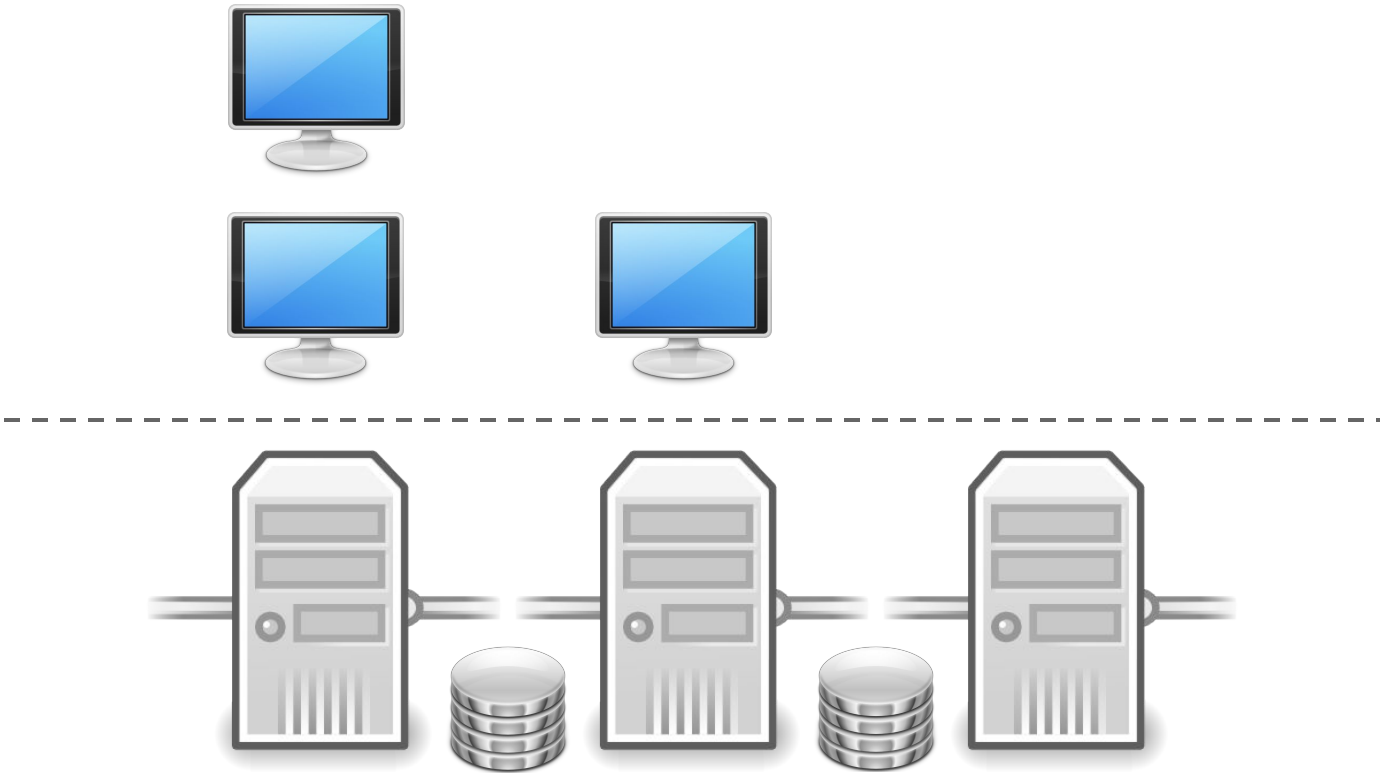
# Modern Use Cases

# Multi-Device Access

# Multi-User Sharing



# Cloud Infrastructure



# Secret-Storage Problem

# Secret-Storage Problem

*How do we store and protect secrets*
*while also supporting*
*a range of modern use cases?*

# Secret-Storage as a Service

# Storage

# Storage

# Access Control

Storage

Access Control

Auditing


LastPass


- Requires single (semi-)trusted third party
- Not designed for automated use cases

LastPass ****

- Requires single (semi-)trusted third party
- Not designed for automated use cases

LastPass ****

- Requires single (semi-)trusted third party
- Not designed for automated use cases


- Lacks support for out-of-band approval
- Designed for single administrative domain

# Tutamen:
## Next-Gen Secret-Storage

# Goals

# Flexible Authentication

Plugins for Multi-factor, Out-of-Band, Etc Auth

## Flexible Authentication

Plugins for Multi-factor, Out-of-Band, Etc Auth

## Minimally Trusted Infrastructure

Sharding Across Multiple Servers

## Flexible Authentication

Plugins for Multi-factor, Out-of-Band, Etc Auth

## Minimally Trusted Infrastructure

Sharding Across Multiple Servers

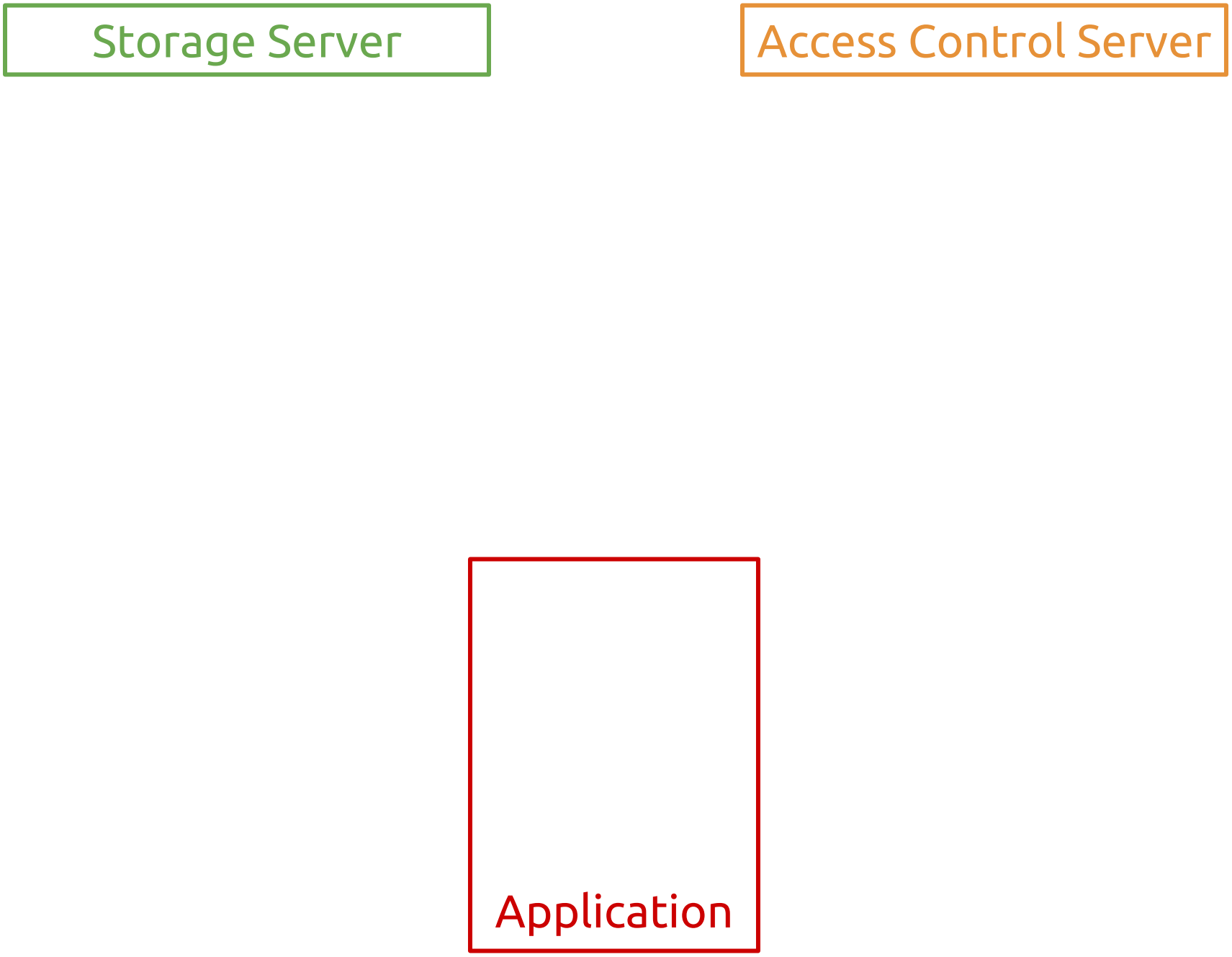
## Beyond a Single Administrative Domain

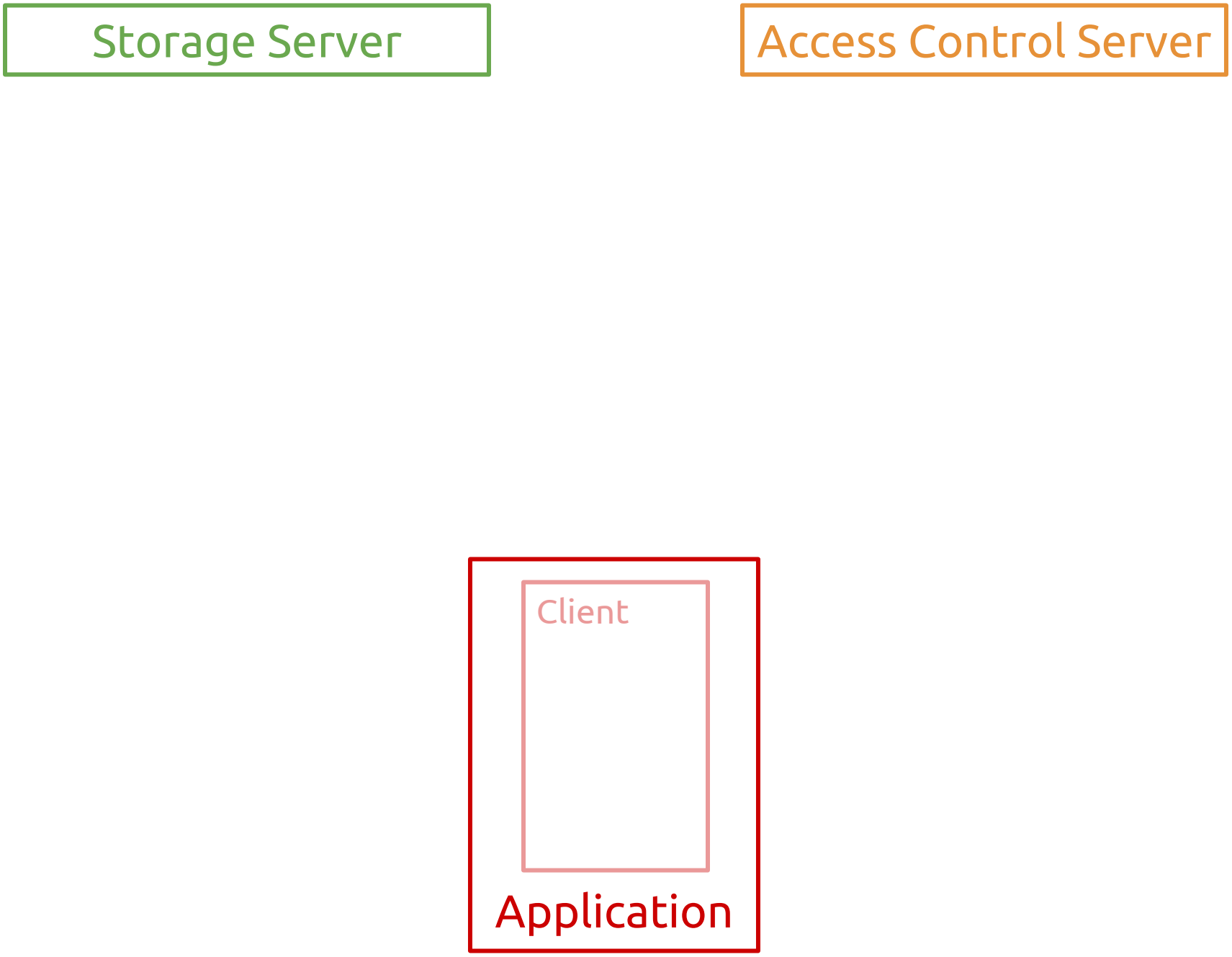
Distributed Federation Between Servers

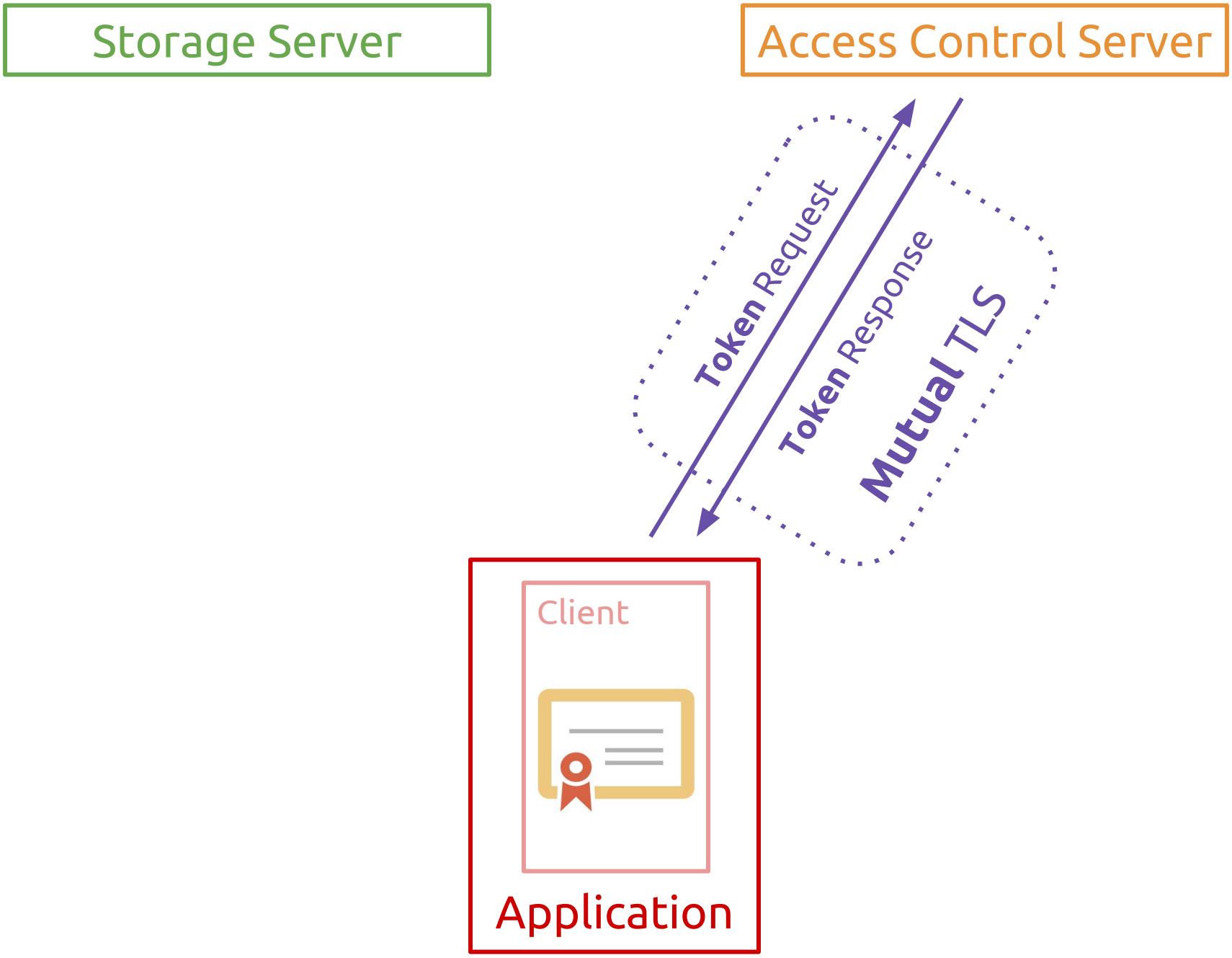
# Architecture

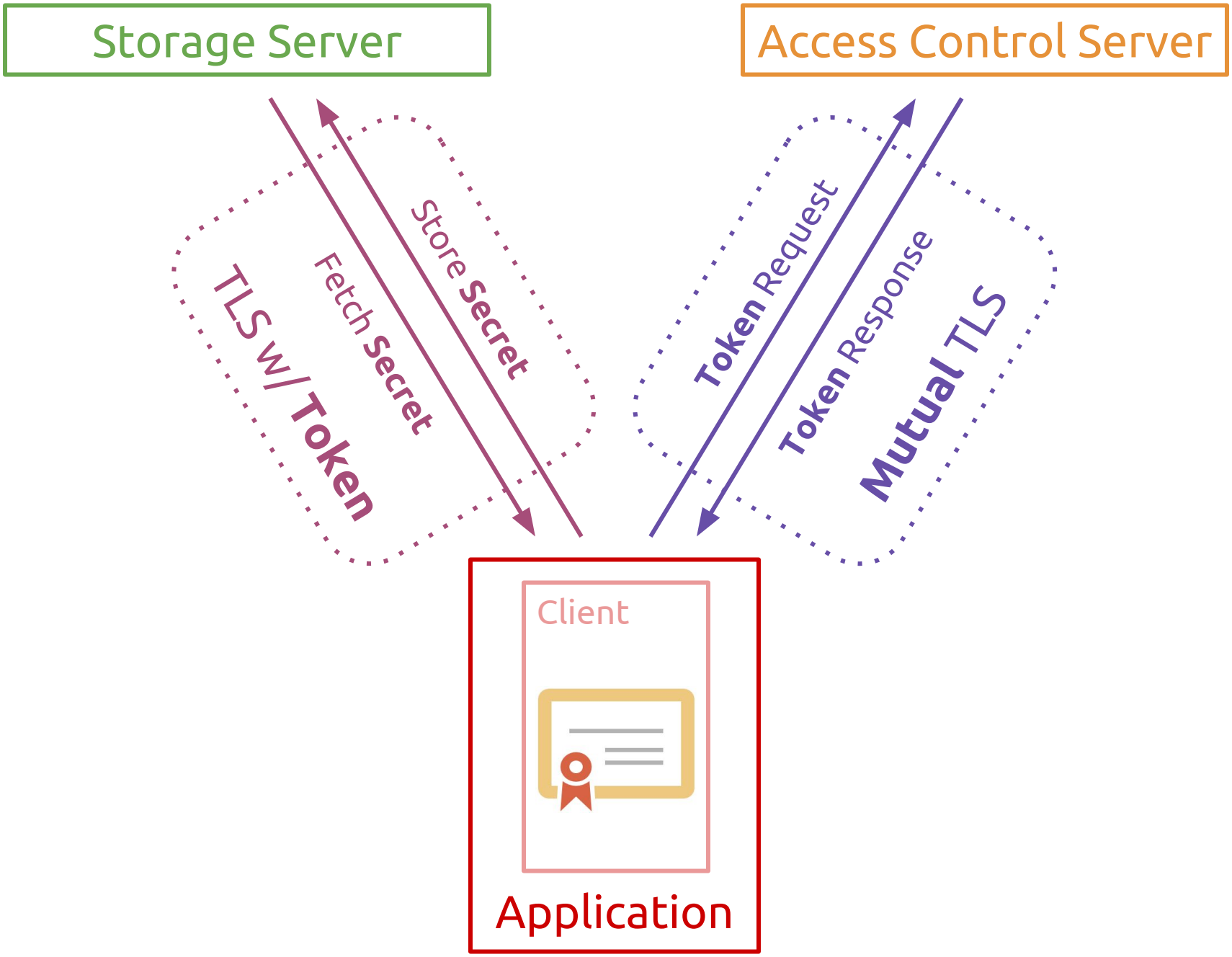

Storage Server

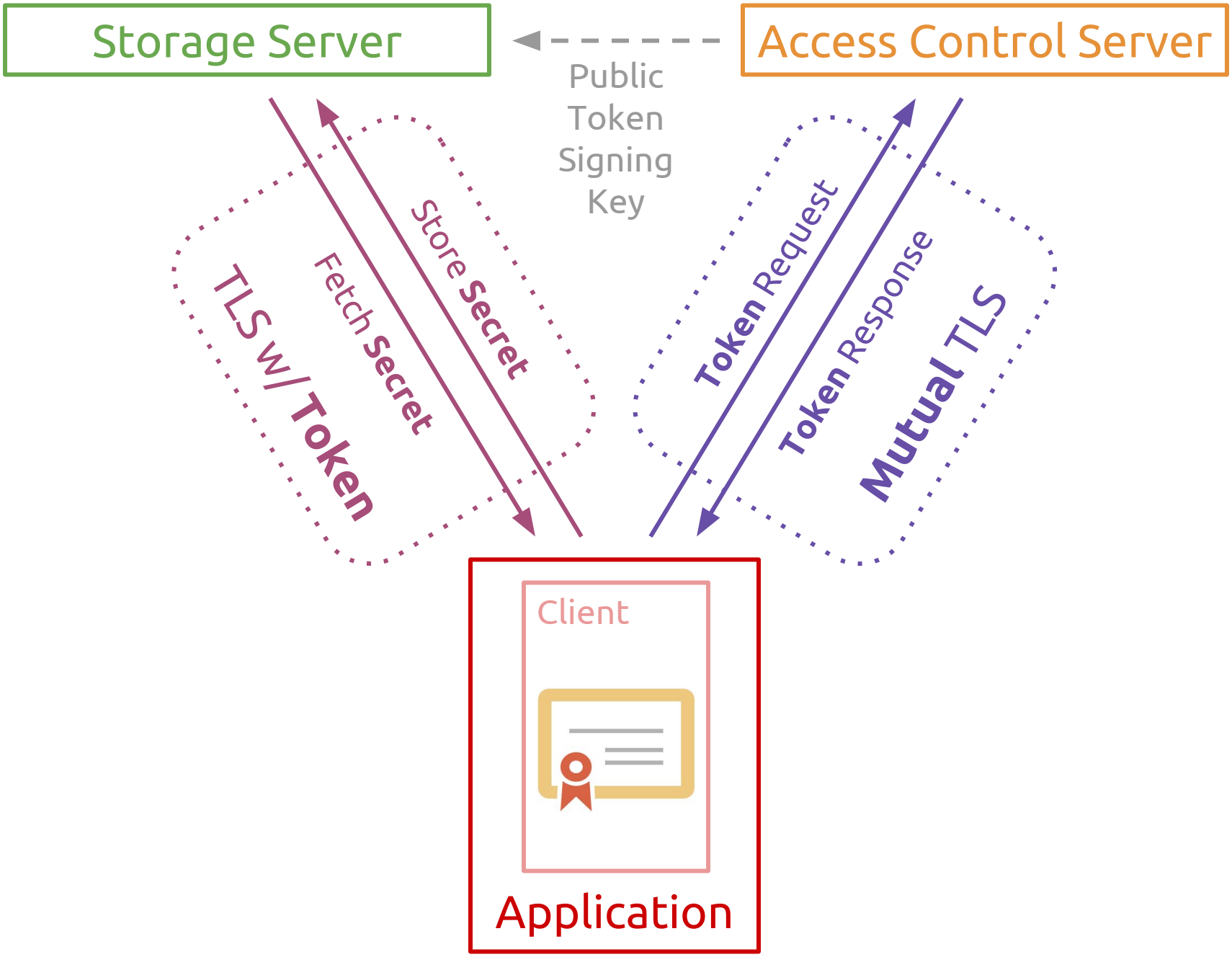

Storage Server
Access Control Server

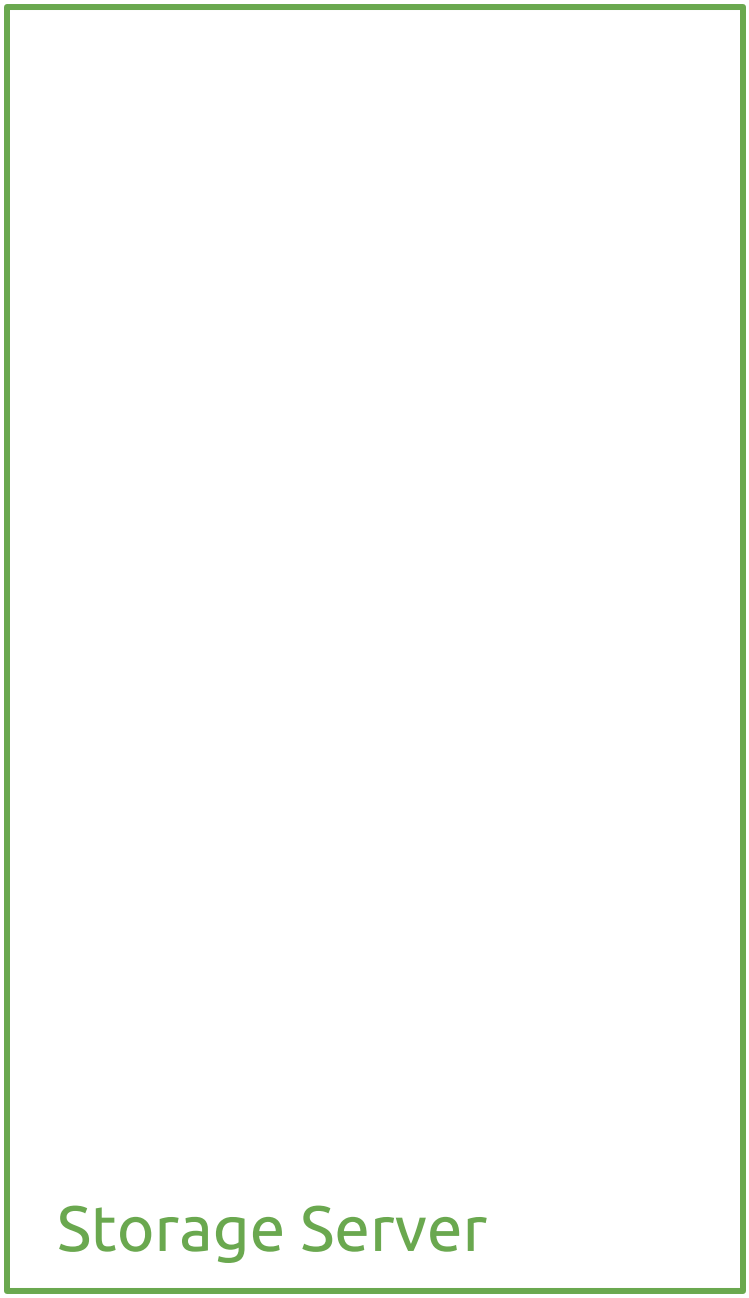

Storage Server
Access Control Server
Application

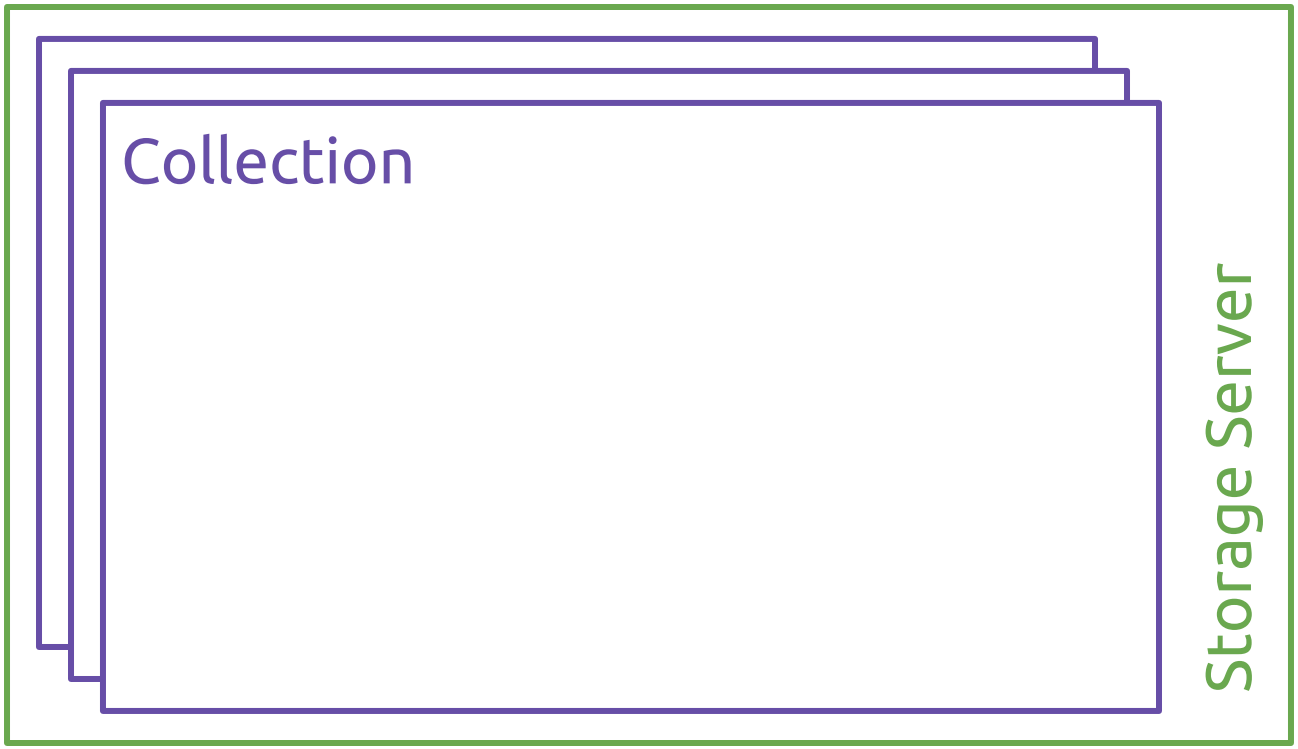

Storage Server
Access Control Server
Client
Application

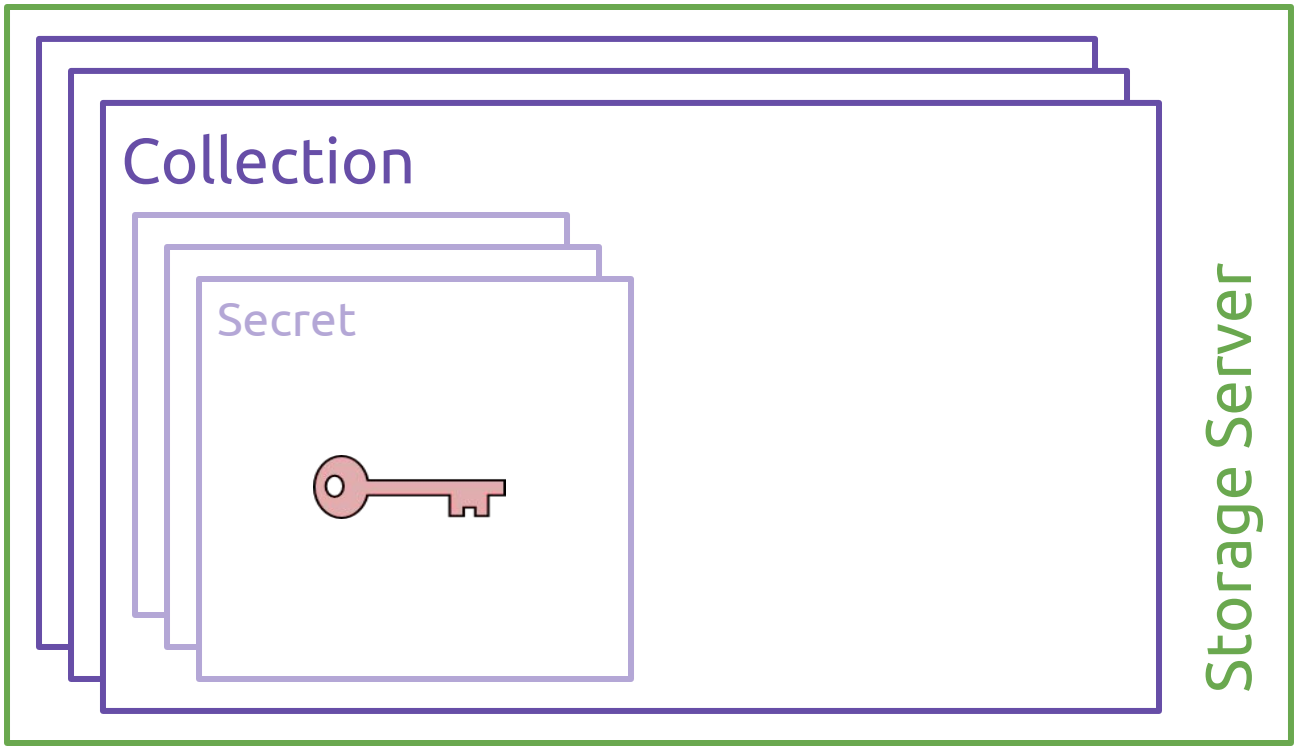

Storage Server
Access Control Server
Token Request
Token Response
Mutual TLS
Client
Application

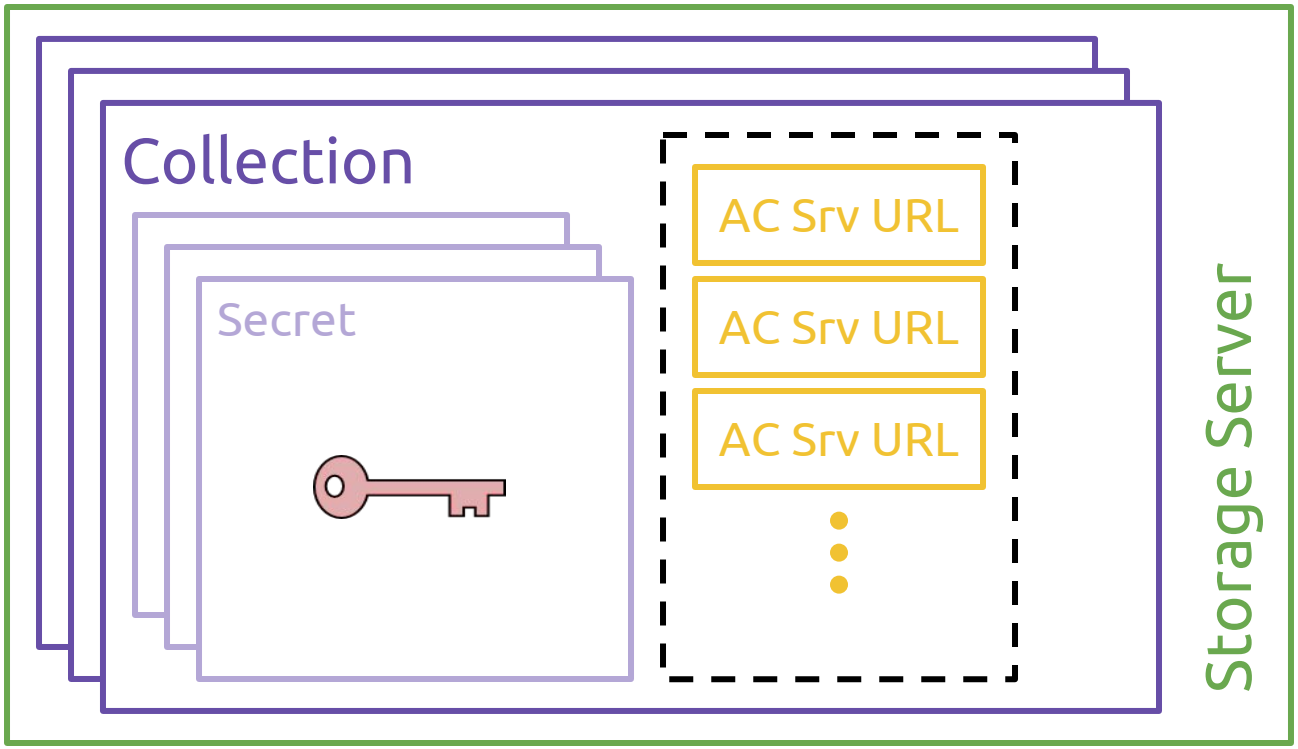

Storage Server
Access Control Server
Store Secret
Fetch Secret
TLS w/ Token
Token Request
Token Response
Mutual TLS
Client
Application

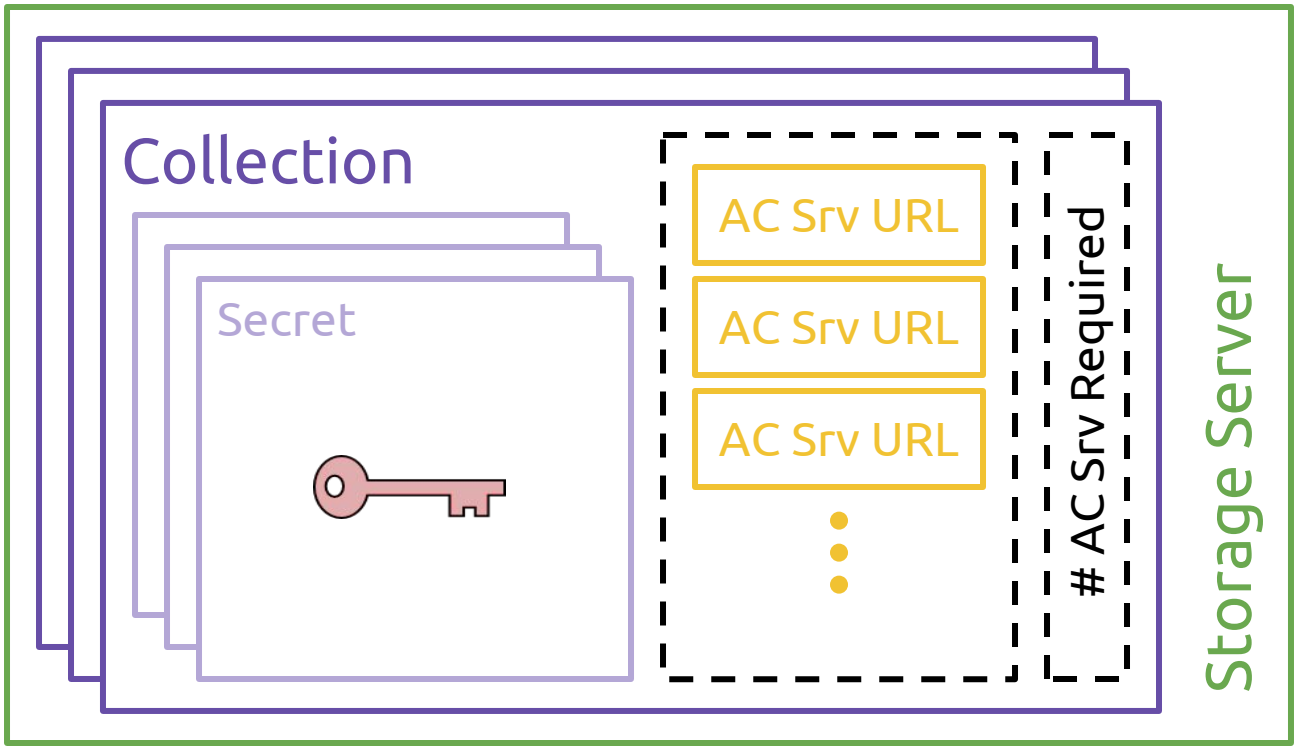

Storage Server
Access Control Server
Public Token Signing Key
Store Secret
Fetch Secret
TLS w/ Token
Token Request
Token Response
Mutual TLS
Client
Application


Storage Server

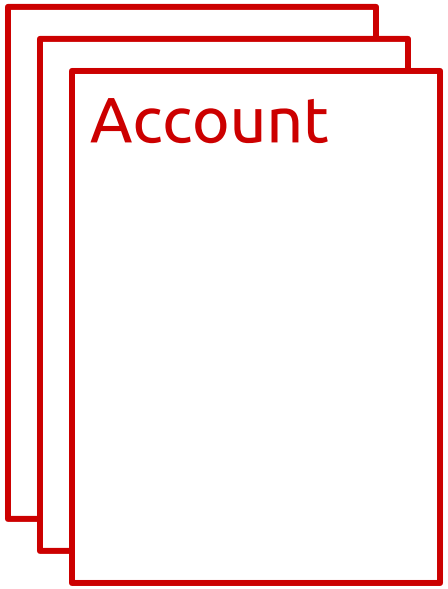

Collection
Storage Server


Collection
Secret
Storage Server

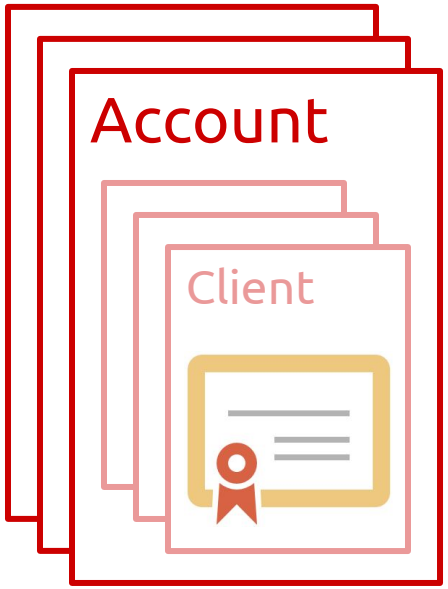

Collection
Secret
AC Srv URL
AC Srv URL
AC Srv URL
Storage Server


Collection
Secret
AC Srv URL
AC Srv URL
AC Srv URL
# AC Srv Required
Storage Server

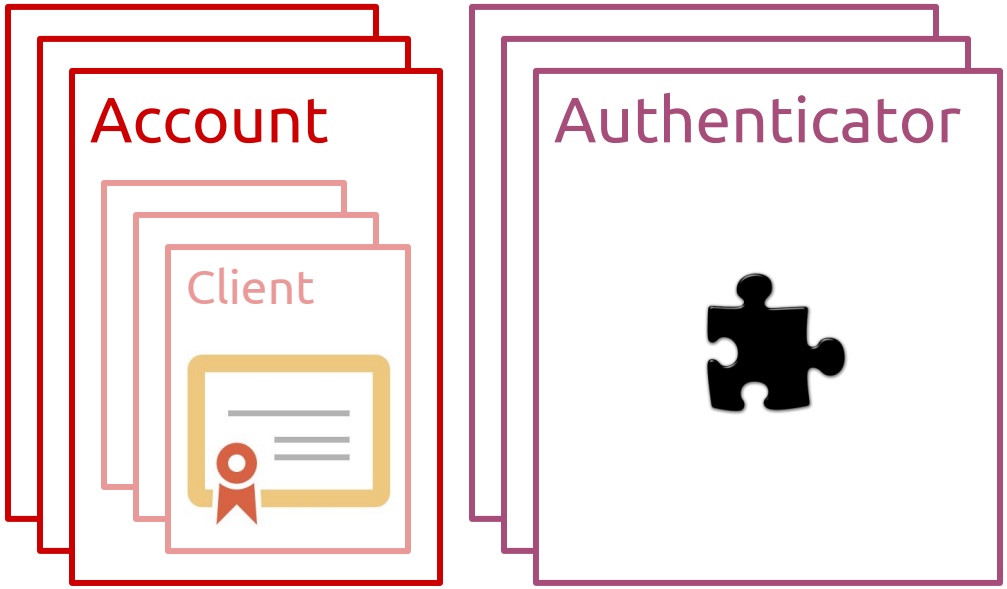
# Access Control Server



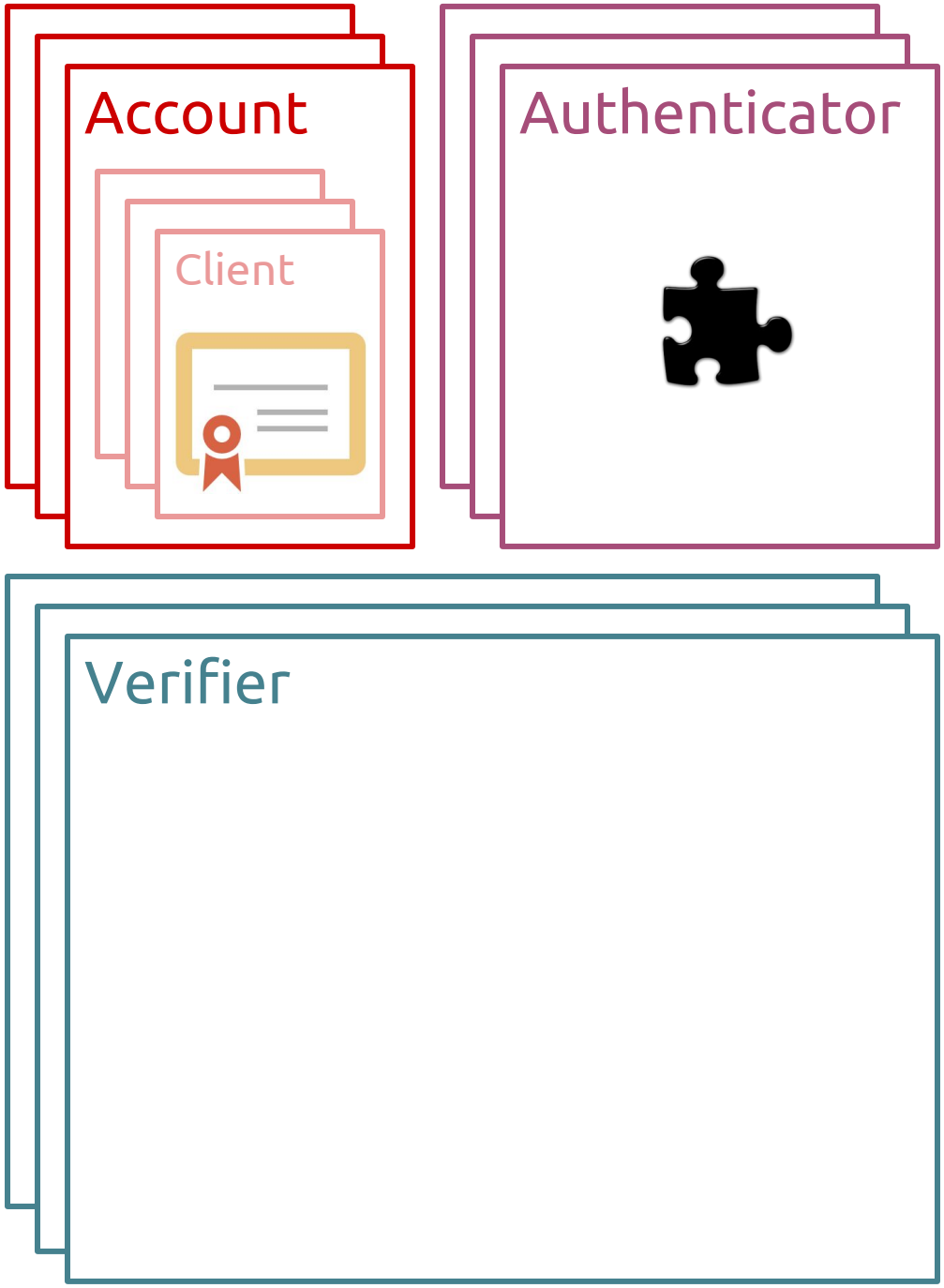

Access Control Server

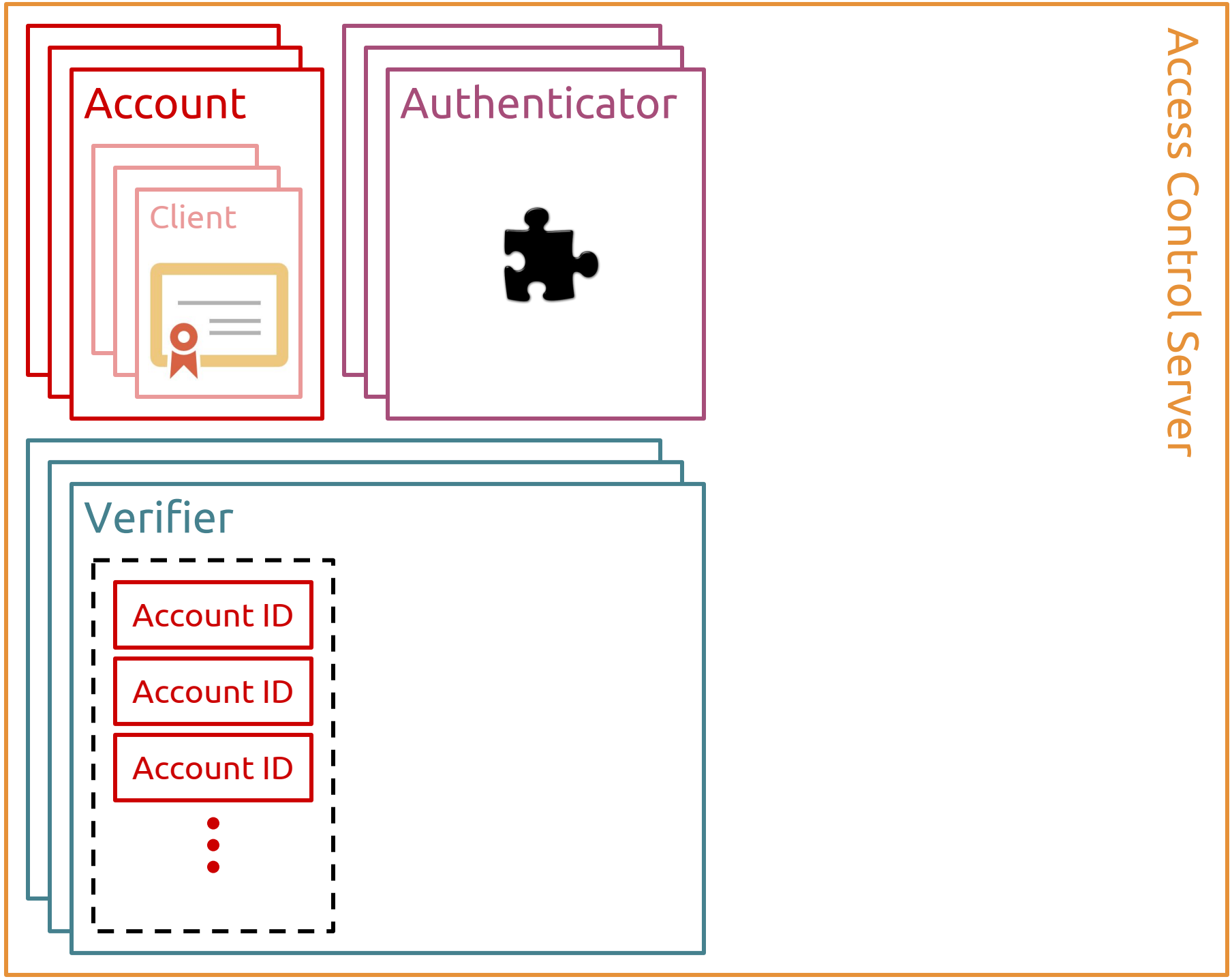


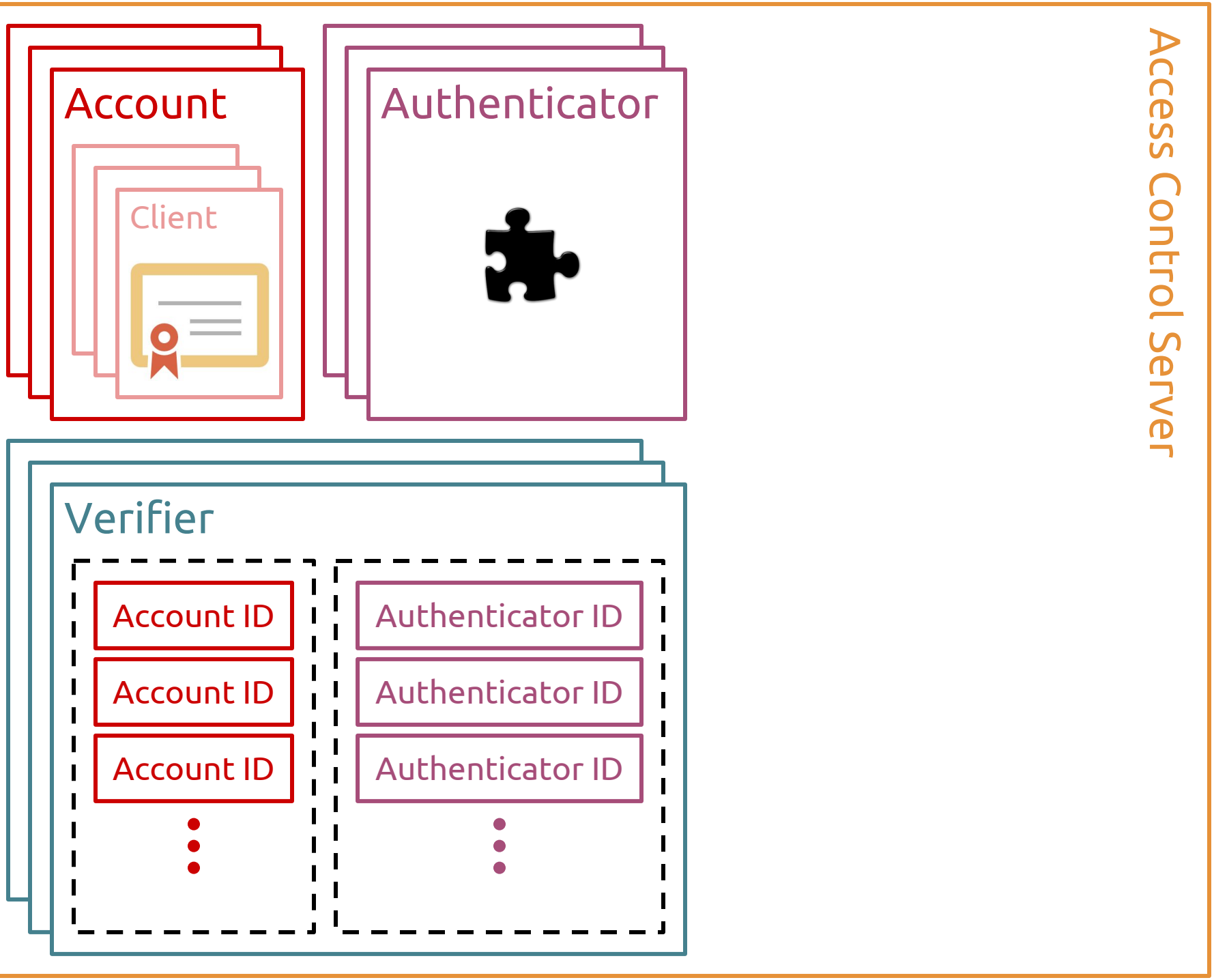

Access Control Server

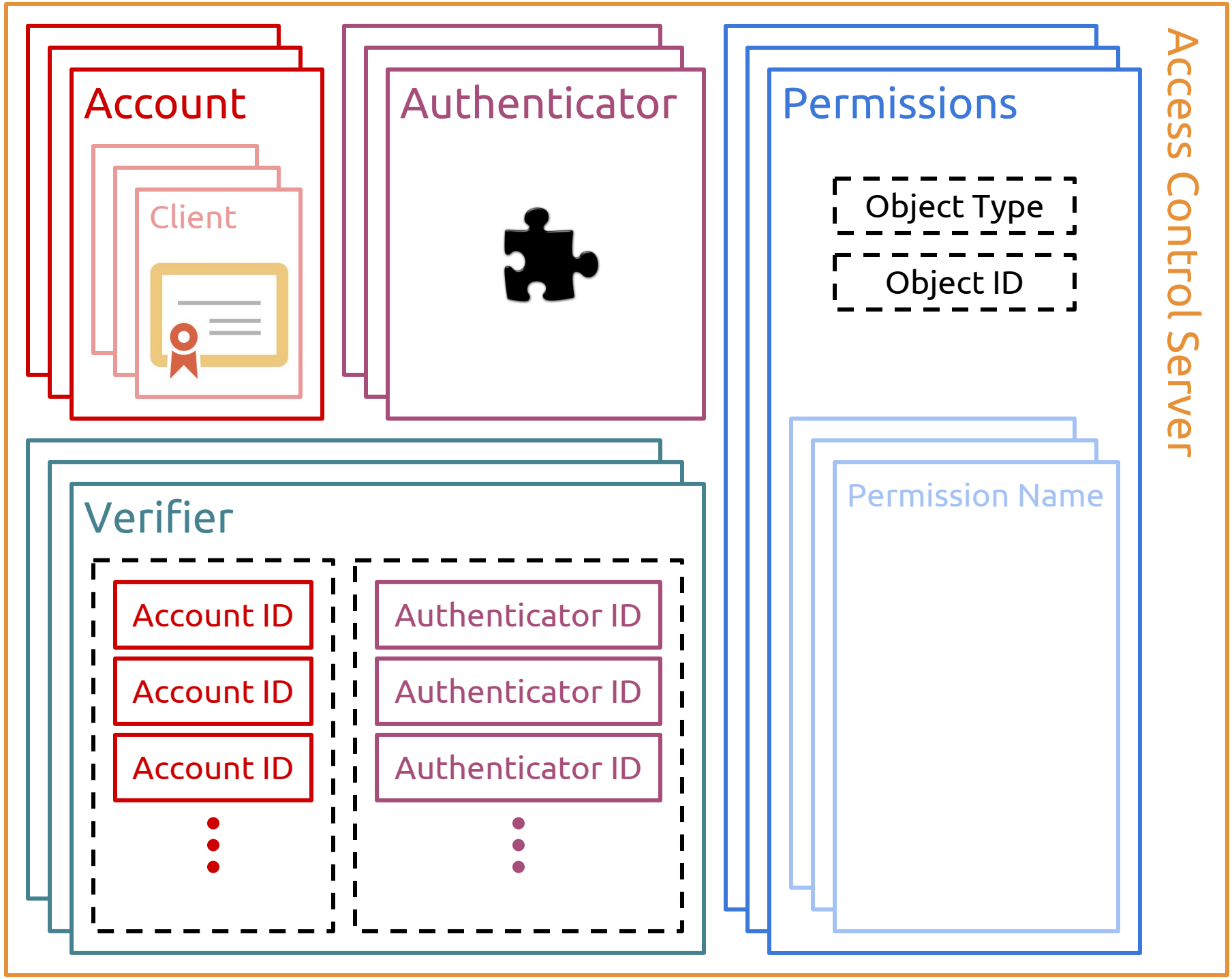


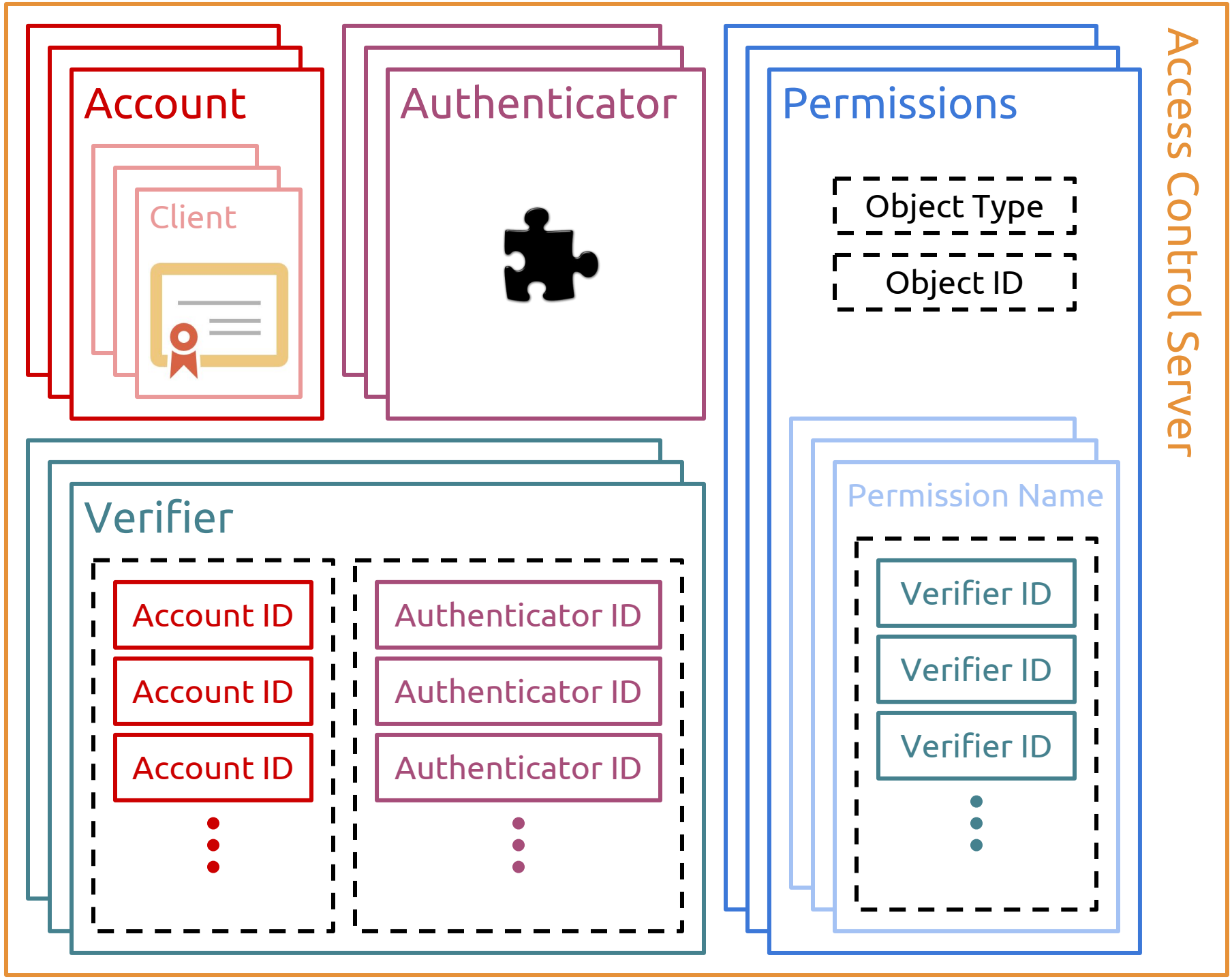

Access Control Server


Account
Client
Authenticator
Verifier
Access Control Server


Account
Client
Authenticator
Verifier
Account ID
Account ID
Account ID
Access Control Server

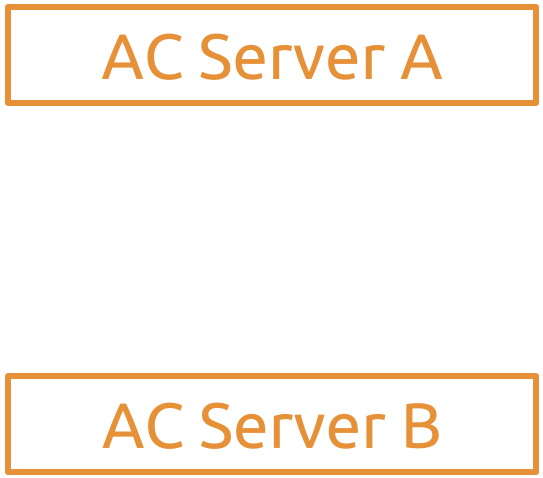

Account
Client
Authenticator
Verifier
Account ID
Account ID
Account ID
Authenticator ID
Authenticator ID
Authenticator ID
Access Control Server

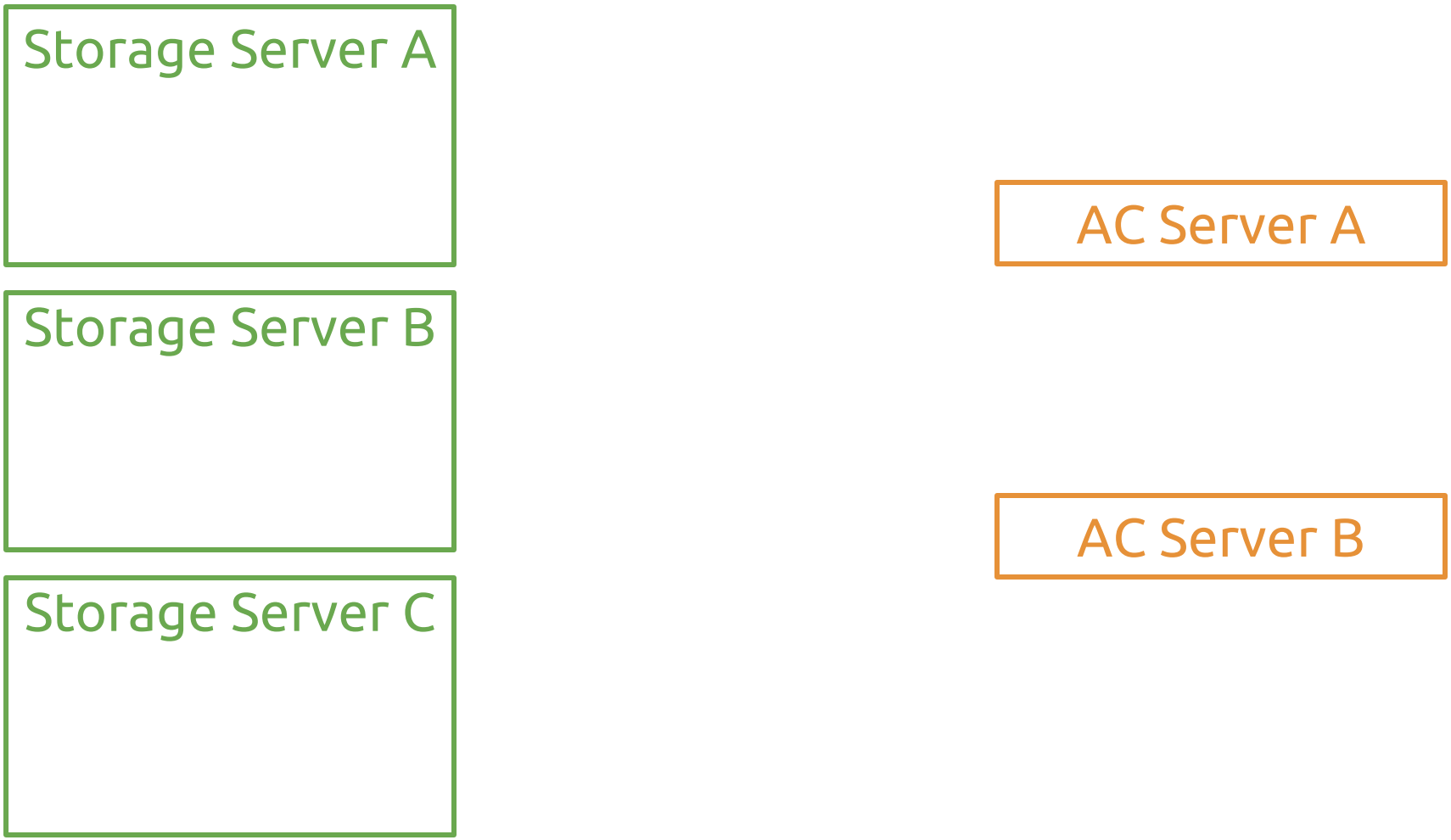

Account
Client
Authenticator
Permissions
Object Type
Object ID
Permission Name
Verifier
Account ID
Account ID
Account ID
Authenticator ID
Authenticator ID
Authenticator ID
Access Control Server


Access Control Server
Account
Client
Authenticator
Permissions
Object Type
Object ID
Permission Name
Verifier ID
Verifier ID
Verifier ID
Verifier
Account ID
Account ID
Account ID
Authenticator ID
Authenticator ID
Authenticator ID

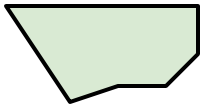
# Why Place Trust In Single Servers?

# Multi-Server Operation

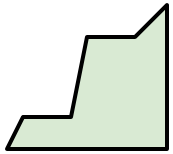

AC Server A
AC Server B


Storage Server A
Storage Server B
Storage Server C
AC Server A
AC Server B

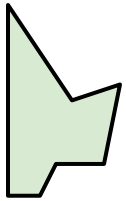
Storage Server A



Storage Server B



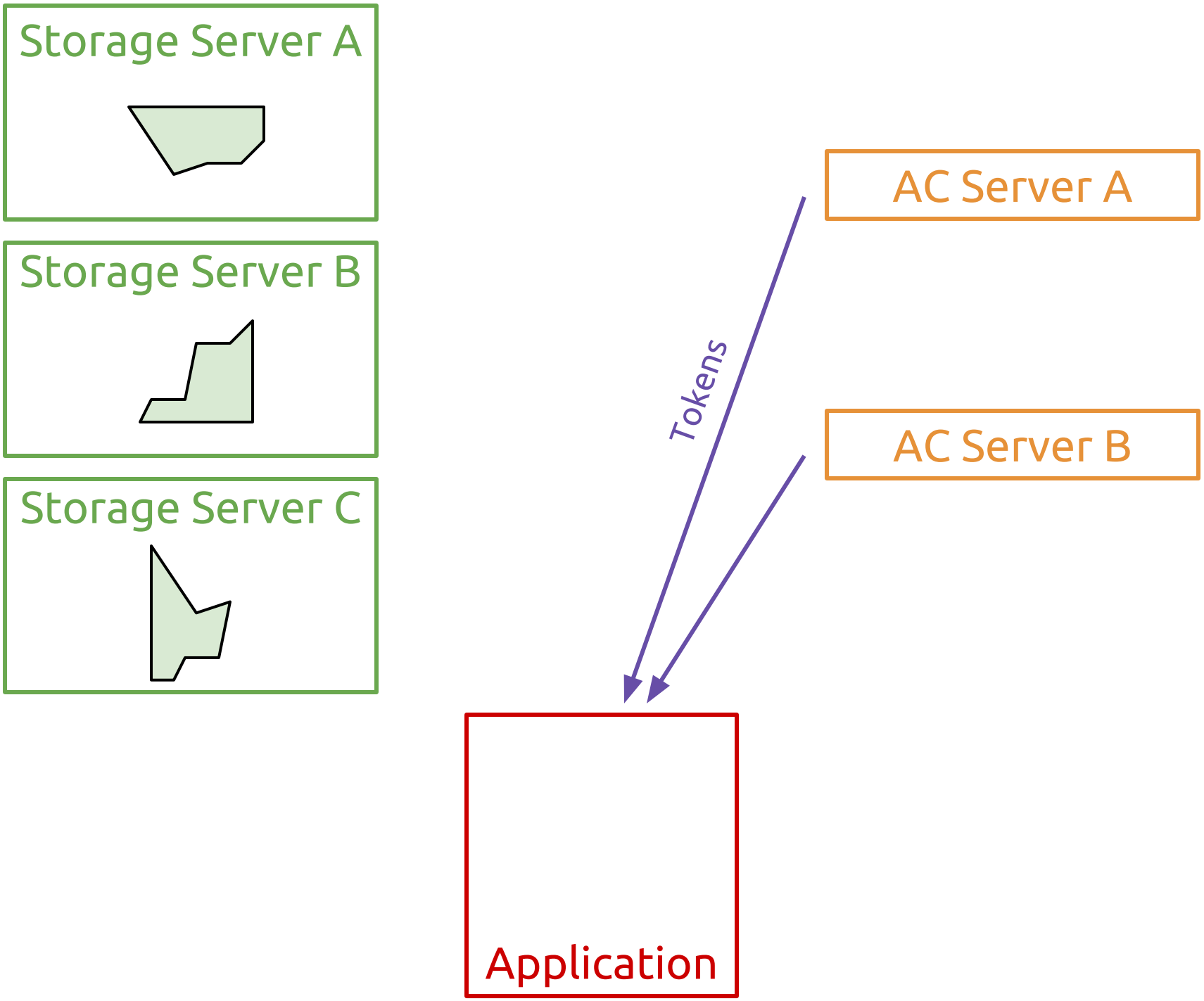
Storage Server C

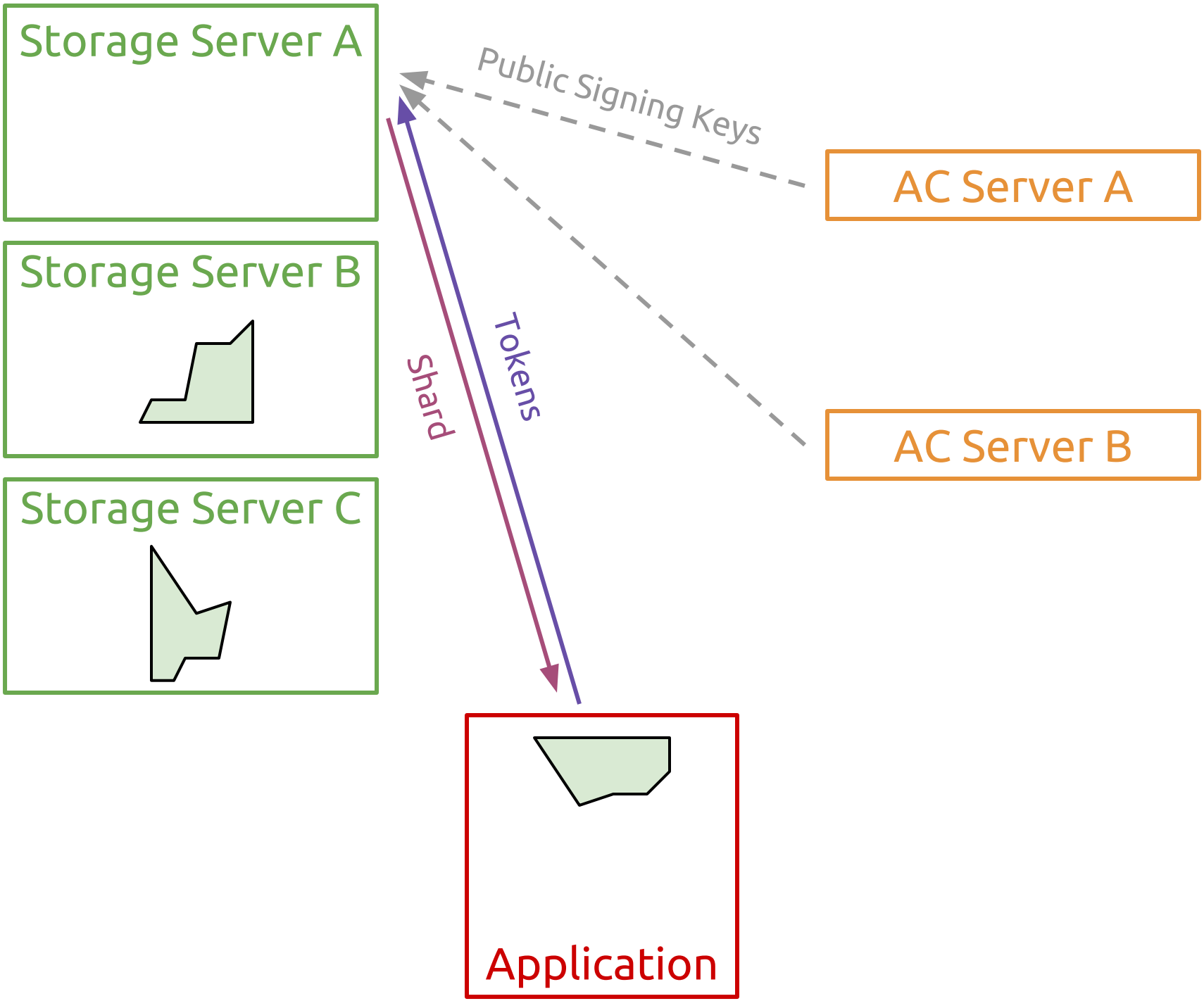


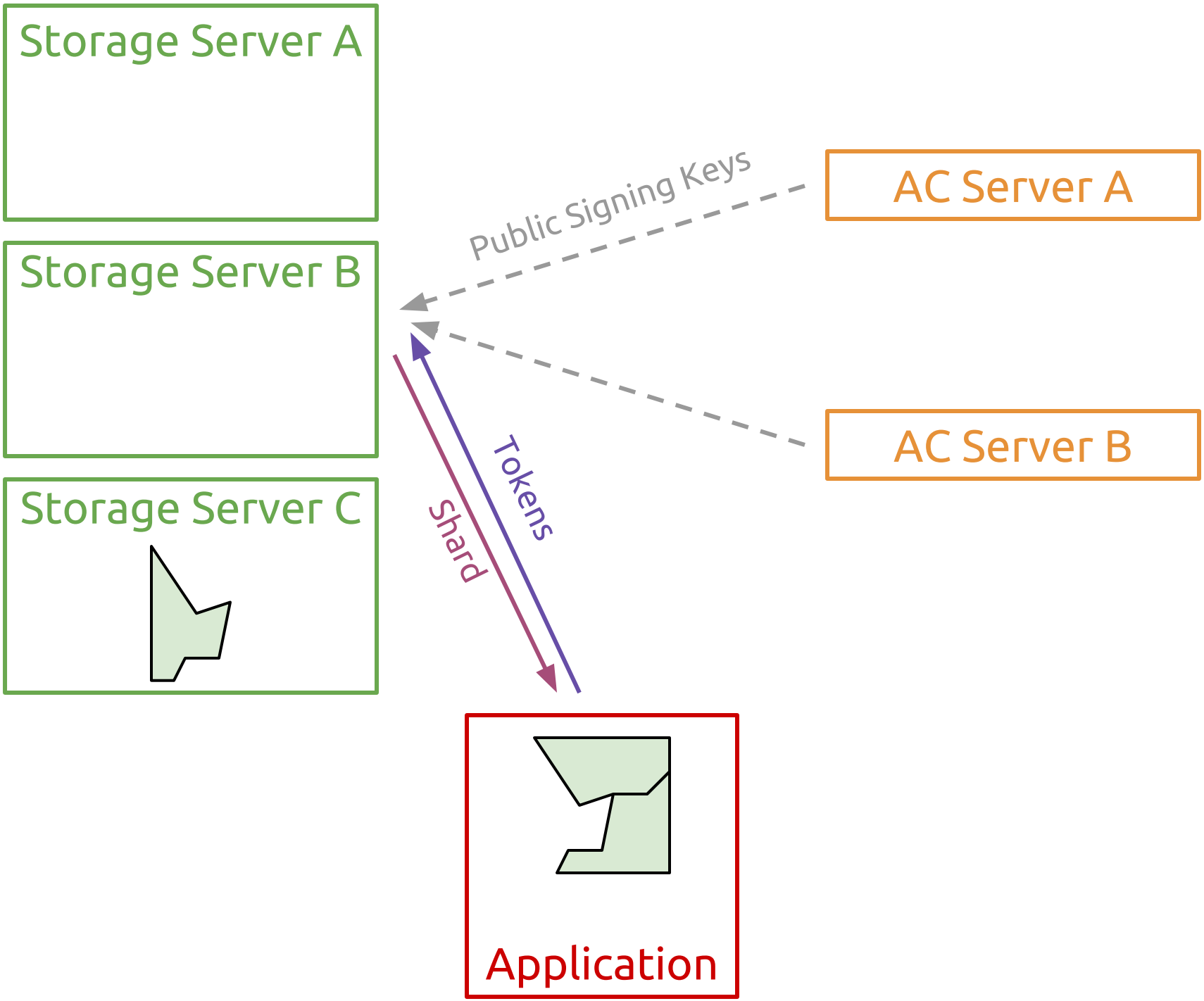
AC Server A

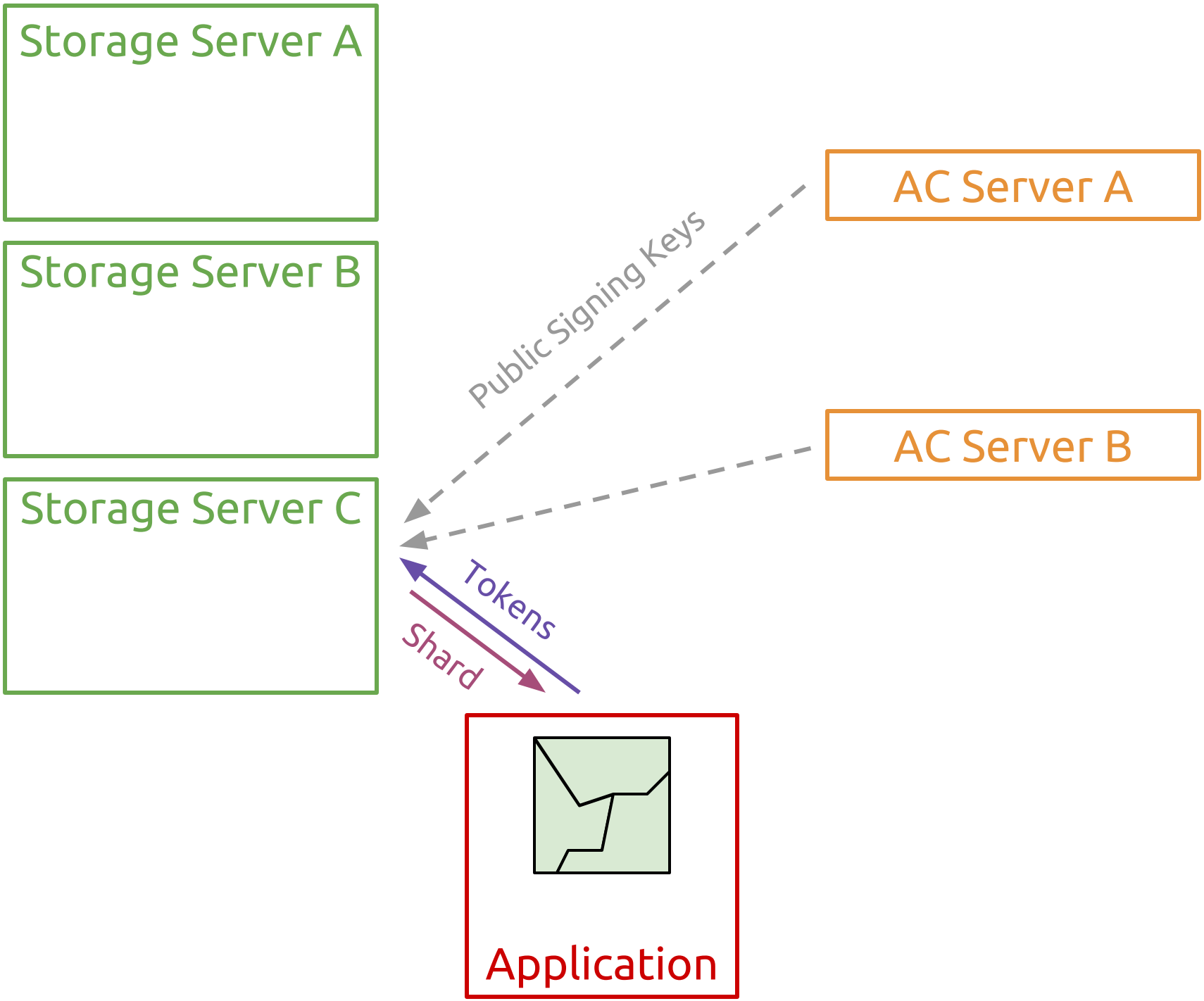
AC Server B

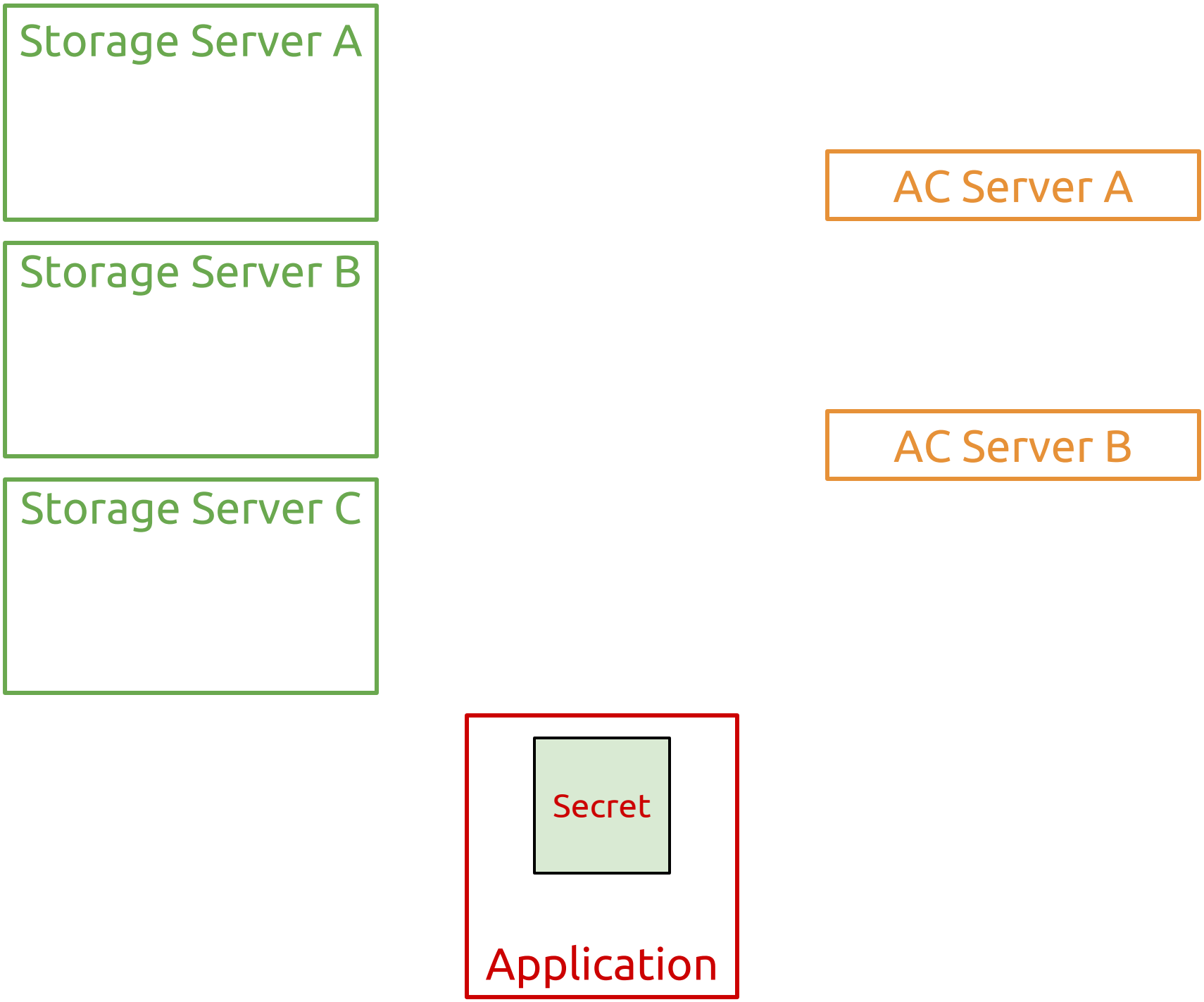
Application


Storage Server A
Storage Server B
Storage Server C
AC Server A
AC Server B
Tokens
Application


Storage Server A
Storage Server B
Storage Server C
Public Signing Keys
AC Server A
AC Server B
Shard
Tokens
Application


Storage Server A
Storage Server B
Storage Server C
Public Signing Keys
AC Server A
AC Server B
Tokens
Shard
Application


Storage Server A
Storage Server B
Storage Server C
AC Server A
AC Server B
Public Signing Keys
Tokens
Shard
Application


Storage Server A
Storage Server B
Storage Server C
AC Server A
AC Server B
Secret
Application

# Secret Retrieval

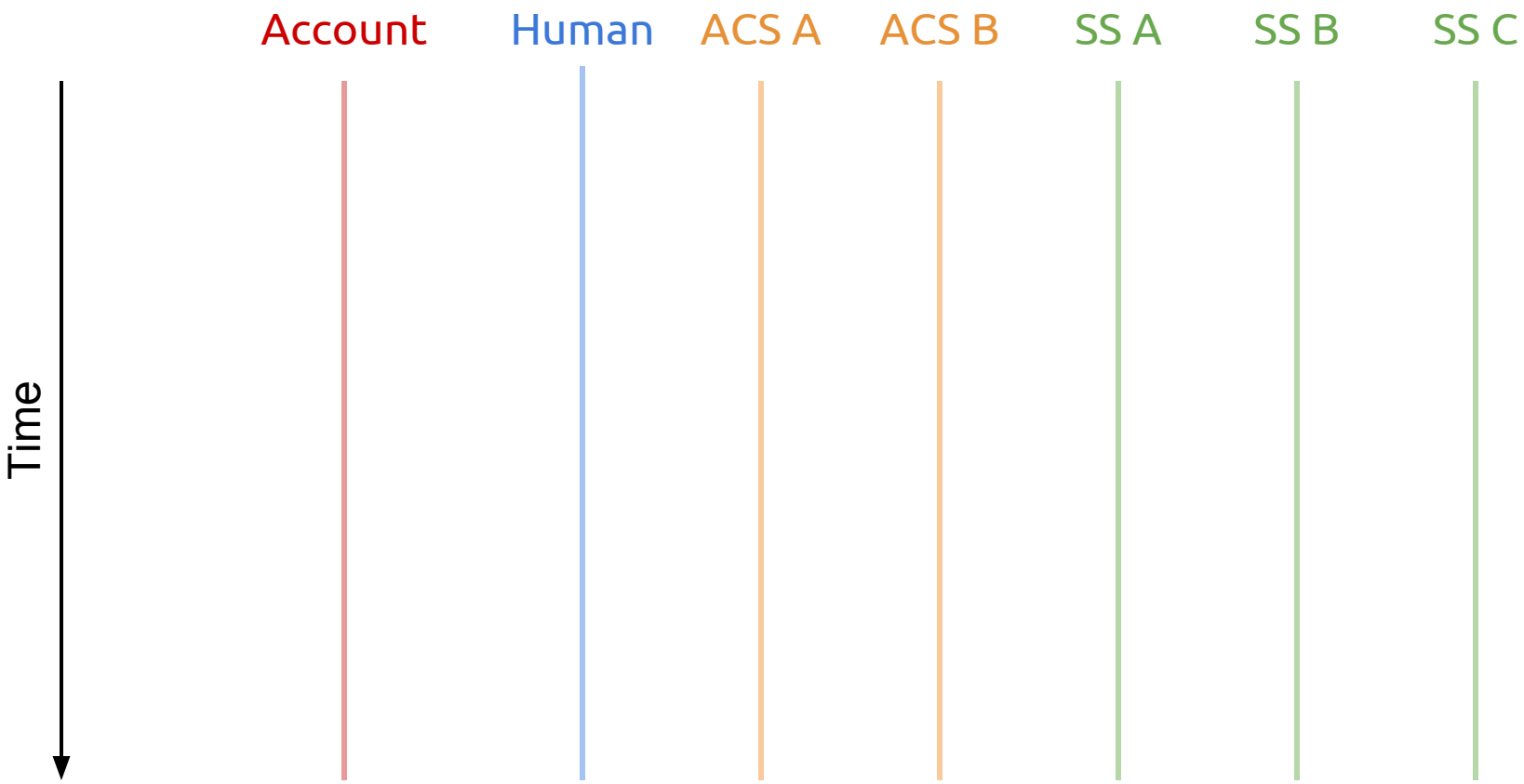
# Secret Retrieval

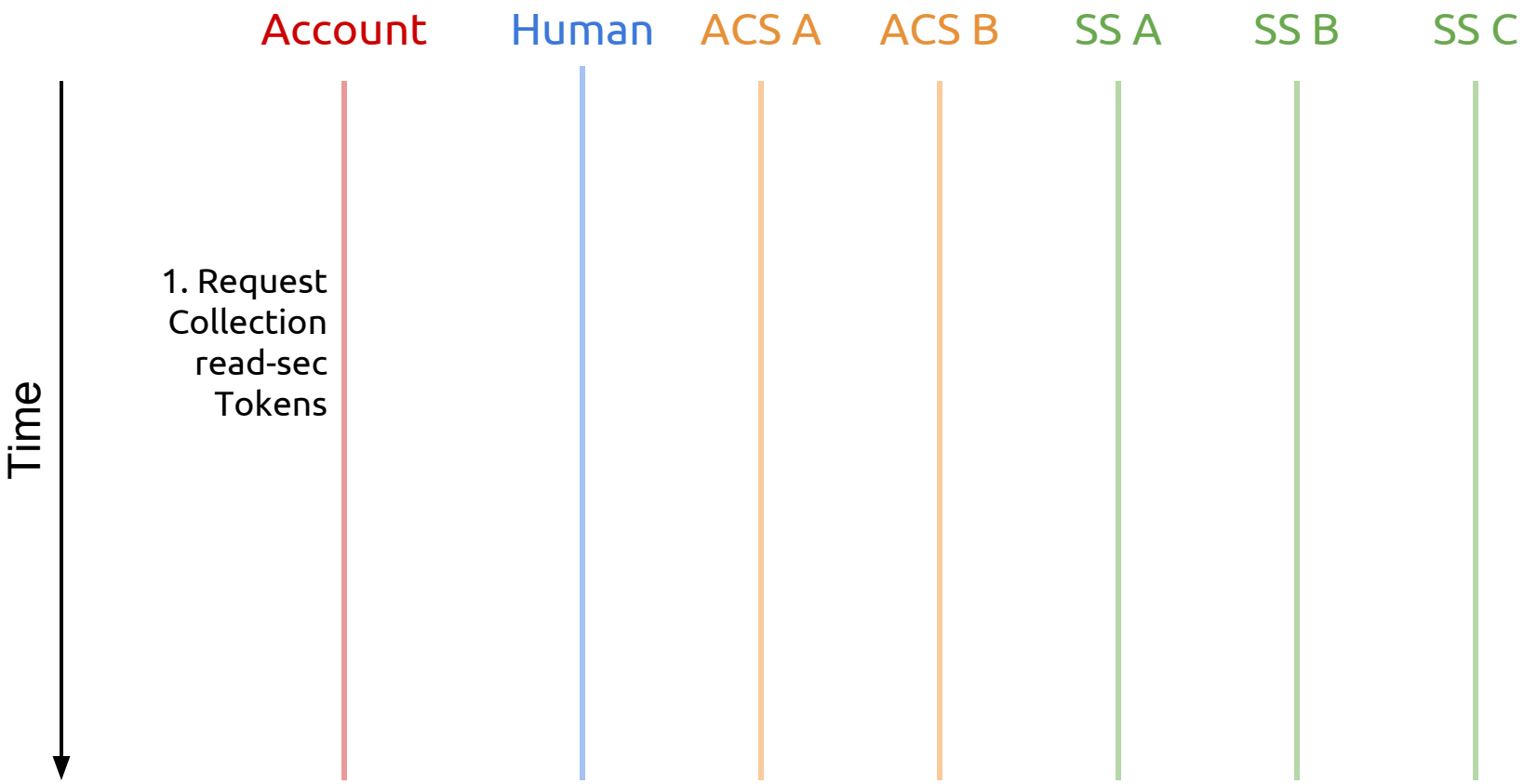
## w/ Out of Band Human-in-the-Loop

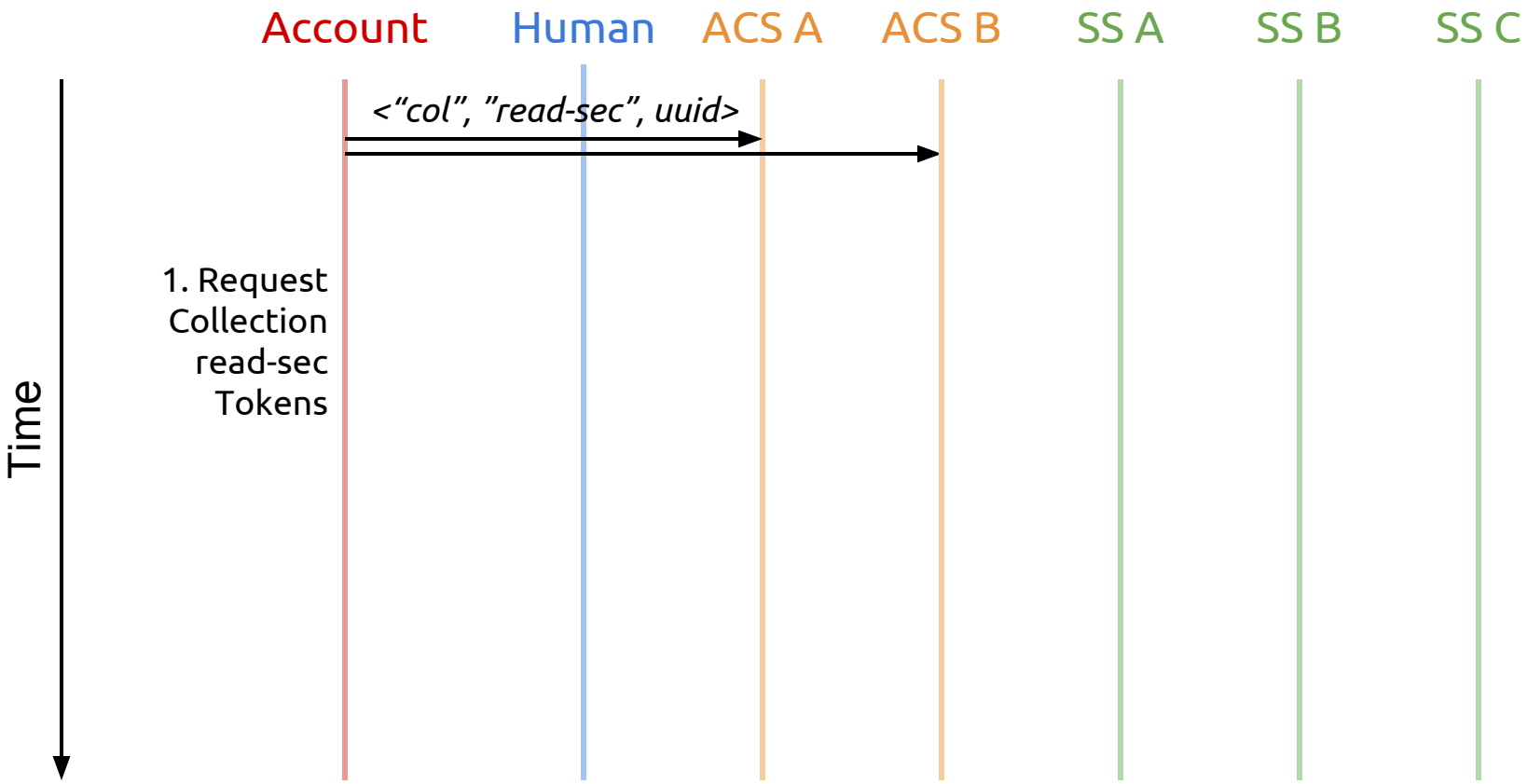
```
Permissions for Collection cf3529eb13be:
  { read: [ Verifier a74b2e2d493d ] }
```

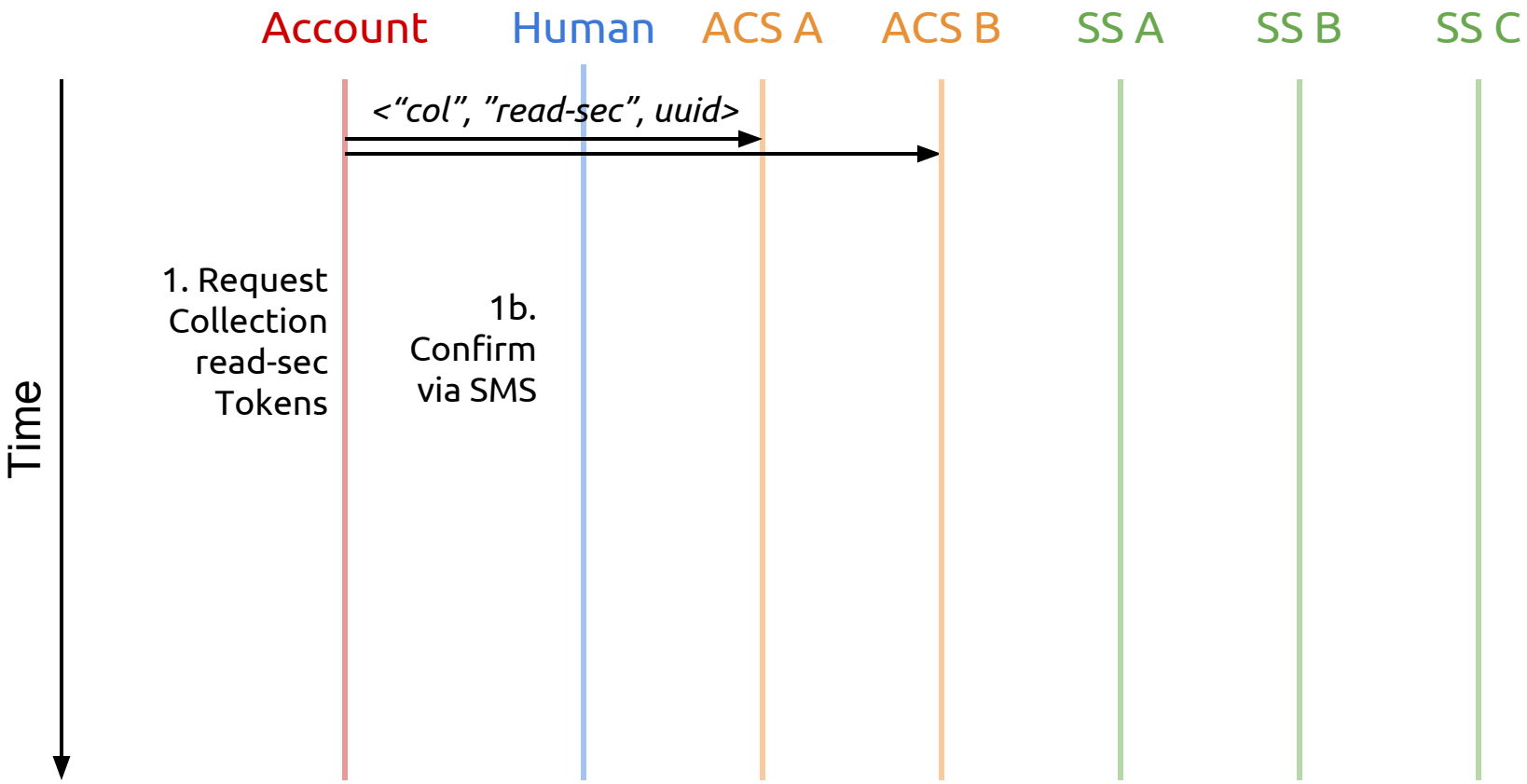
```
Permissions for Collection cf3529eb13be:
  { read: [ Verifier a74b2e2d493d ] }

Verifier a74b2e2d493d
  { Accounts: [ Account cceb832edcdb ] }
    Authenticators: [ Authenticator 34e85e1bb264 ] }
```

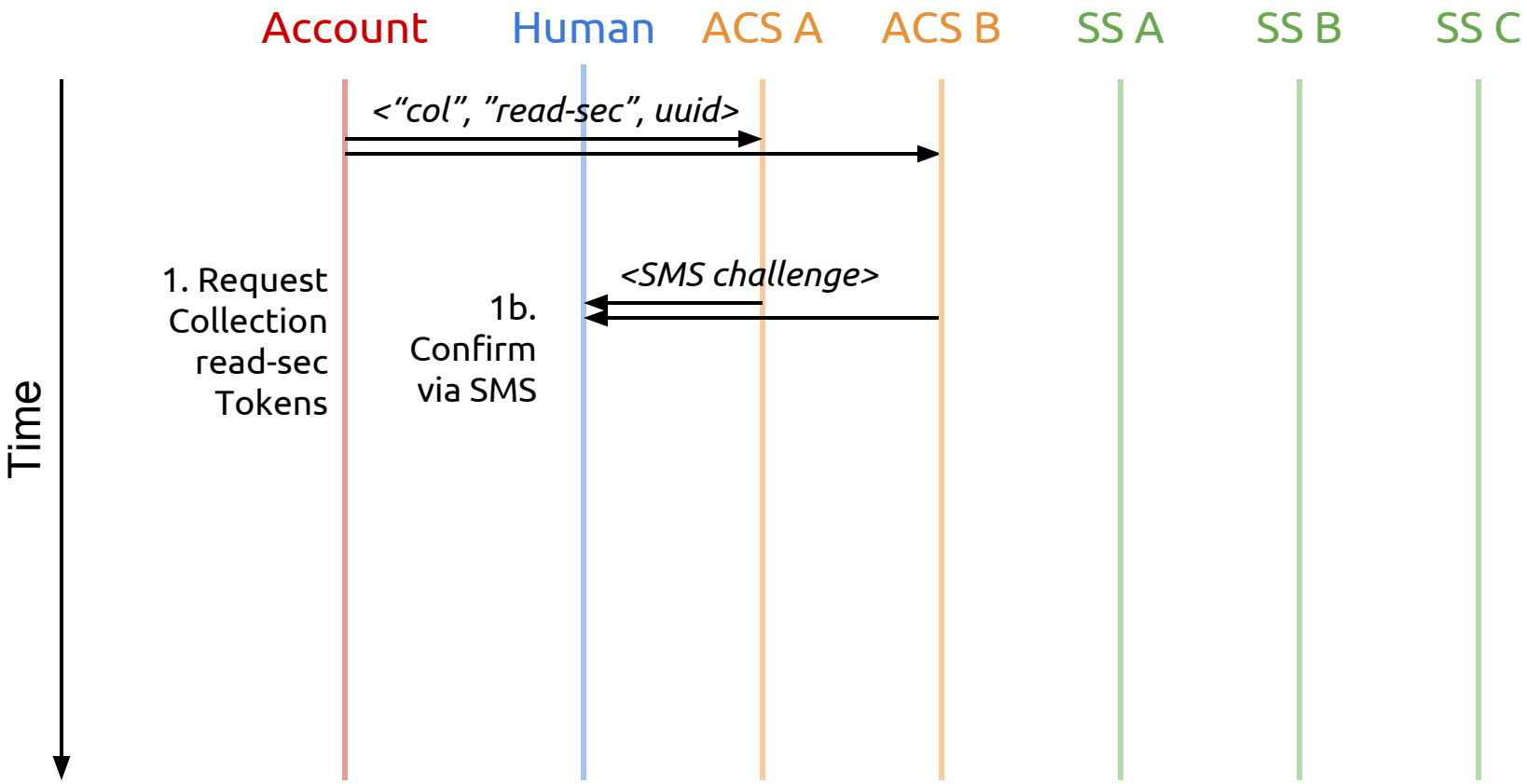
```
Permissions for Collection cf3529eb13be:
  { read: [ Verifier a74b2e2d493d ] }

Verifier a74b2e2d493d
  { Accounts: [ Account cceb832edcdb ] }
    Authenticators: [ Authenticator 34e85e1bb264 ] }

Authenticator 34e85e1bb264
  { Plugin: SMS Challenge/Response }
```

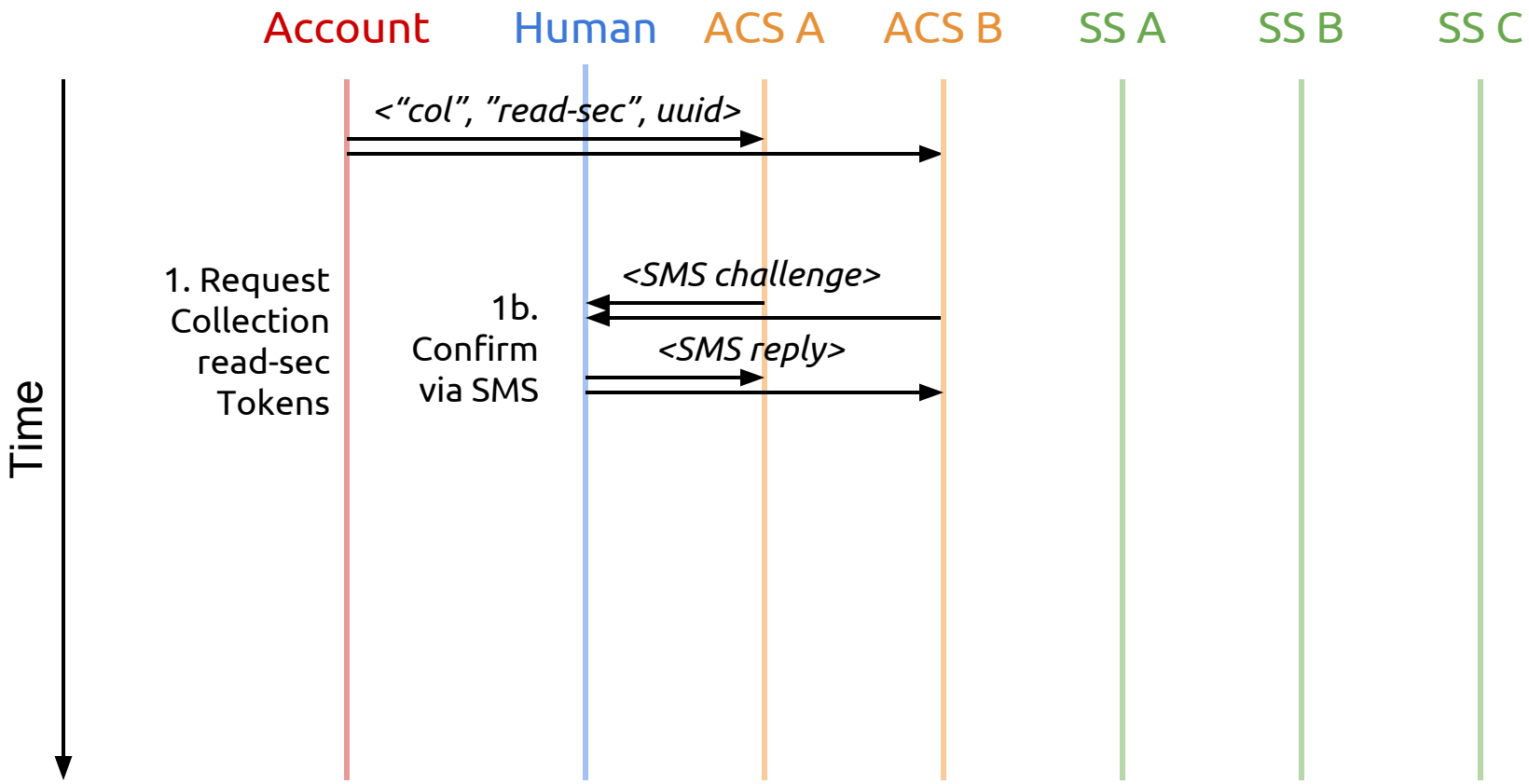

Account
Human
ACS A
ACS B
SS A
SS B
SS C
Time

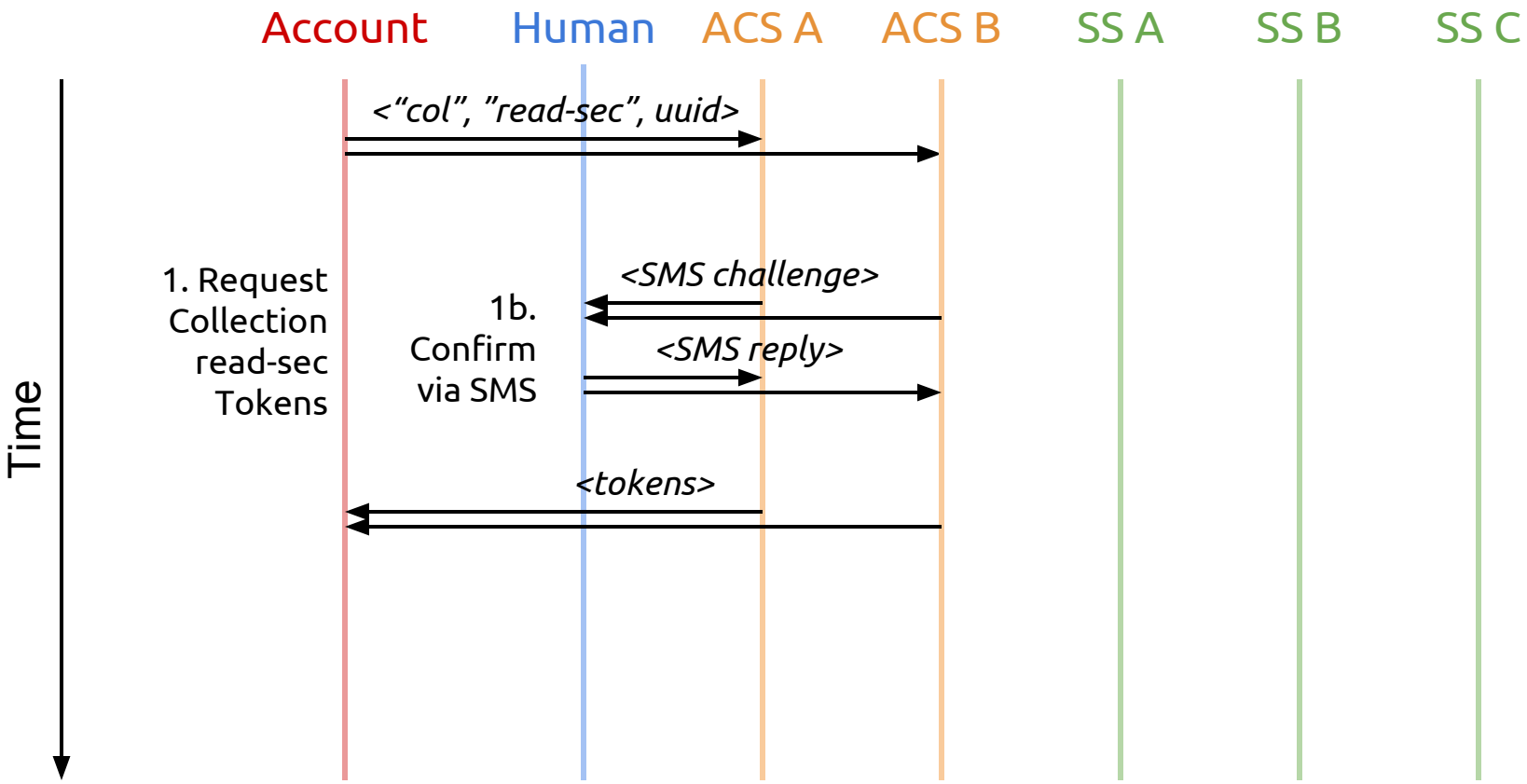

Account
Human
ACS A
ACS B
SS A
SS B
SS C
Time
1. Request
Collection
read-sec
Tokens

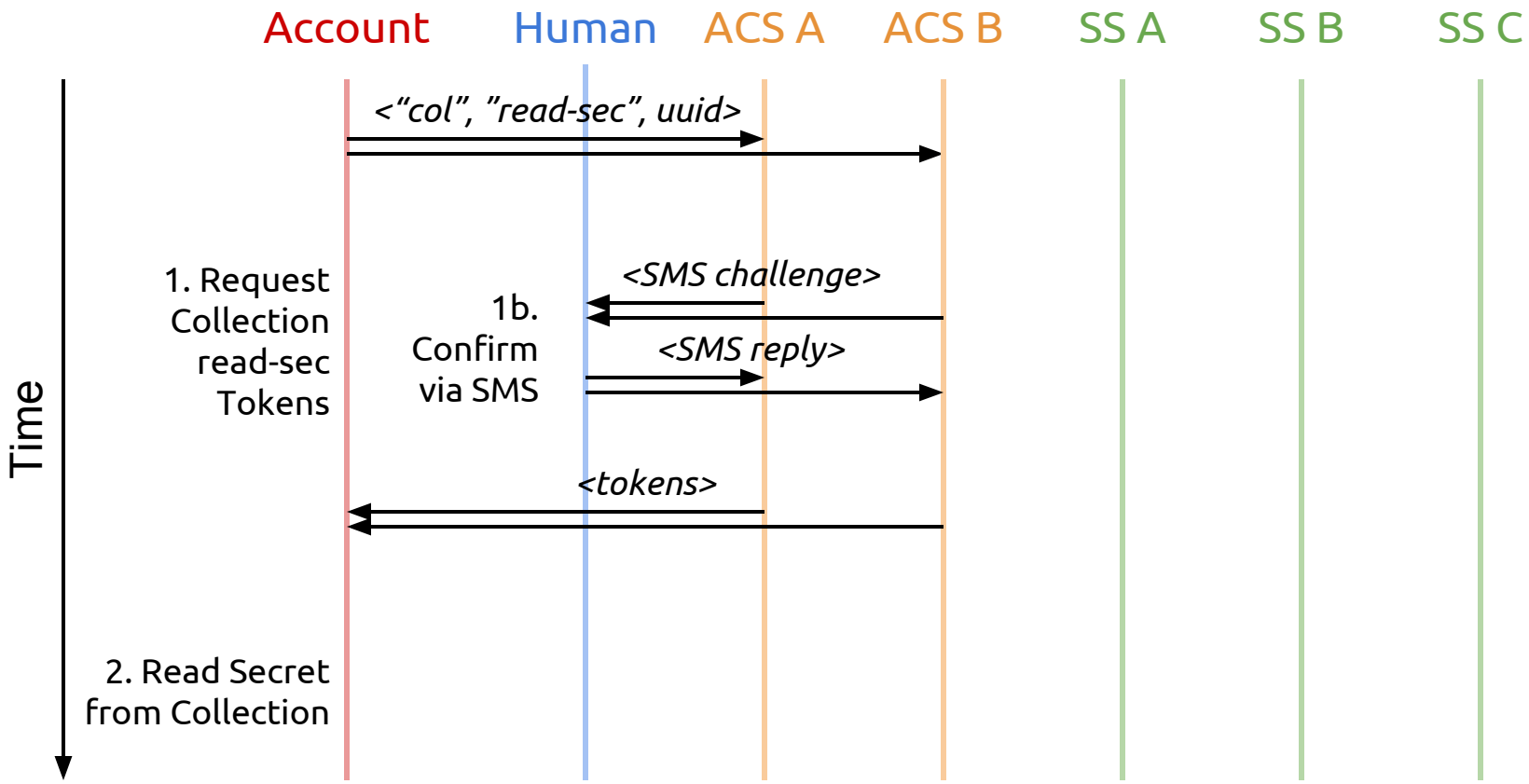

Account
Human
ACS A
ACS B
SS A
SS B
SS C
<"col", "read-sec", uuid>
1. Request
Collection
read-sec
Tokens
Time

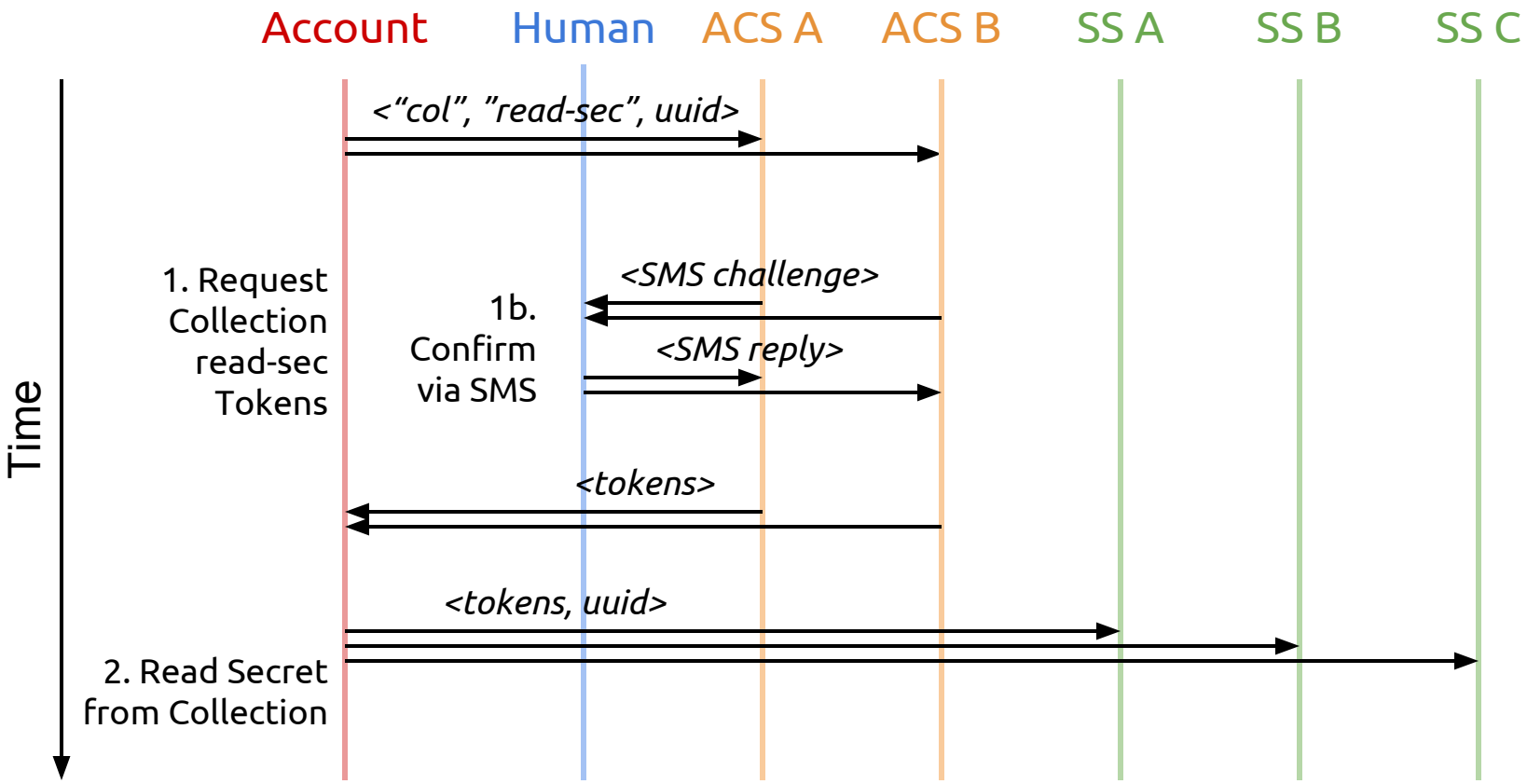

Account
Human
ACS A
ACS B
SS A
SS B
SS C
<"col", "read-sec", uuid>
1. Request Collection read-sec Tokens
1b. Confirm via SMS
Time

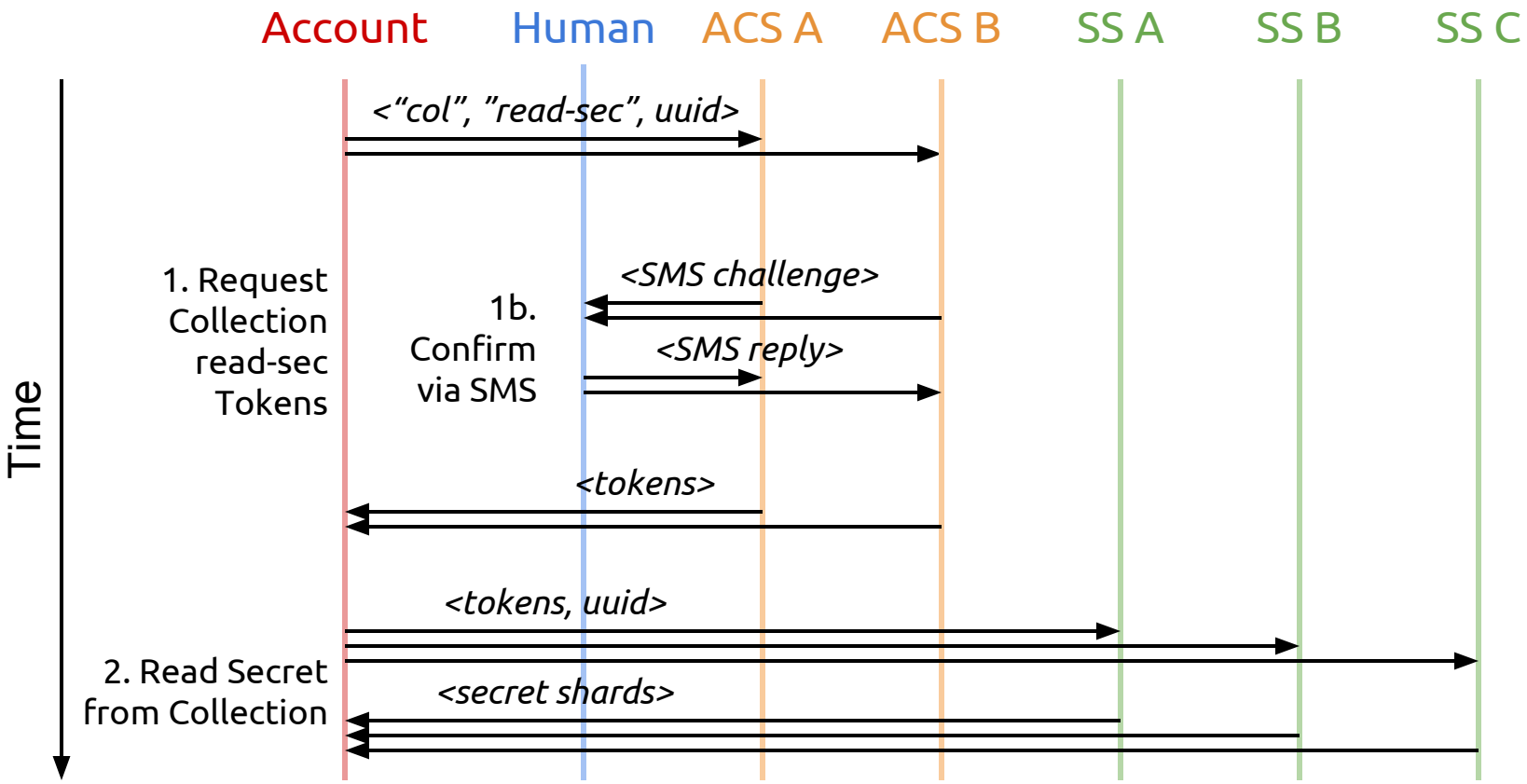

Account
Human
ACS A
ACS B
SS A
SS B
SS C
<"col", "read-sec", uuid>
1. Request
Collection
read-sec
Tokens
<SMS challenge>
1b.
Confirm
via SMS
Time


Account
Human
ACS A
ACS B
SS A
SS B
SS C
Time
<"col", "read-sec", uuid>
1. Request Collection read-sec Tokens
<SMS challenge>
1b. Confirm via SMS
<SMS reply>


Account
Human
ACS A
ACS B
SS A
SS B
SS C
Time
<"col", "read-sec", uuid>
1. Request Collection read-sec Tokens
<SMS challenge>
1b. Confirm via SMS
<SMS reply>
<tokens>


Account
Human
ACS A
ACS B
SS A
SS B
SS C
Time
<"col", "read-sec", uuid>
1. Request Collection read-sec Tokens
<SMS challenge>
1b. Confirm via SMS
<SMS reply>
<tokens>
2. Read Secret from Collection

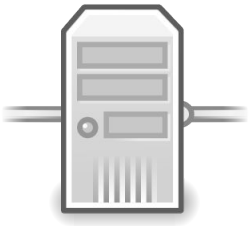

Account
Human
ACS A
ACS B
SS A
SS B
SS C
Time
<"col", "read-sec", uuid>
1. Request Collection read-sec Tokens
<SMS challenge>
1b. Confirm via SMS
<SMS reply>
<tokens>
<tokens, uuid>
2. Read Secret from Collection

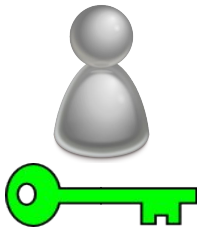

Account
Human
ACS A
ACS B
SS A
SS B
SS C
Time
<"col", "read-sec", uuid>
1. Request Collection read-sec Tokens
<SMS challenge>
1b. Confirm via SMS
<SMS reply>
<tokens>
<tokens, uuid>
2. Read Secret from Collection
<secret shards>

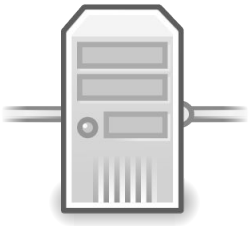
# Applications

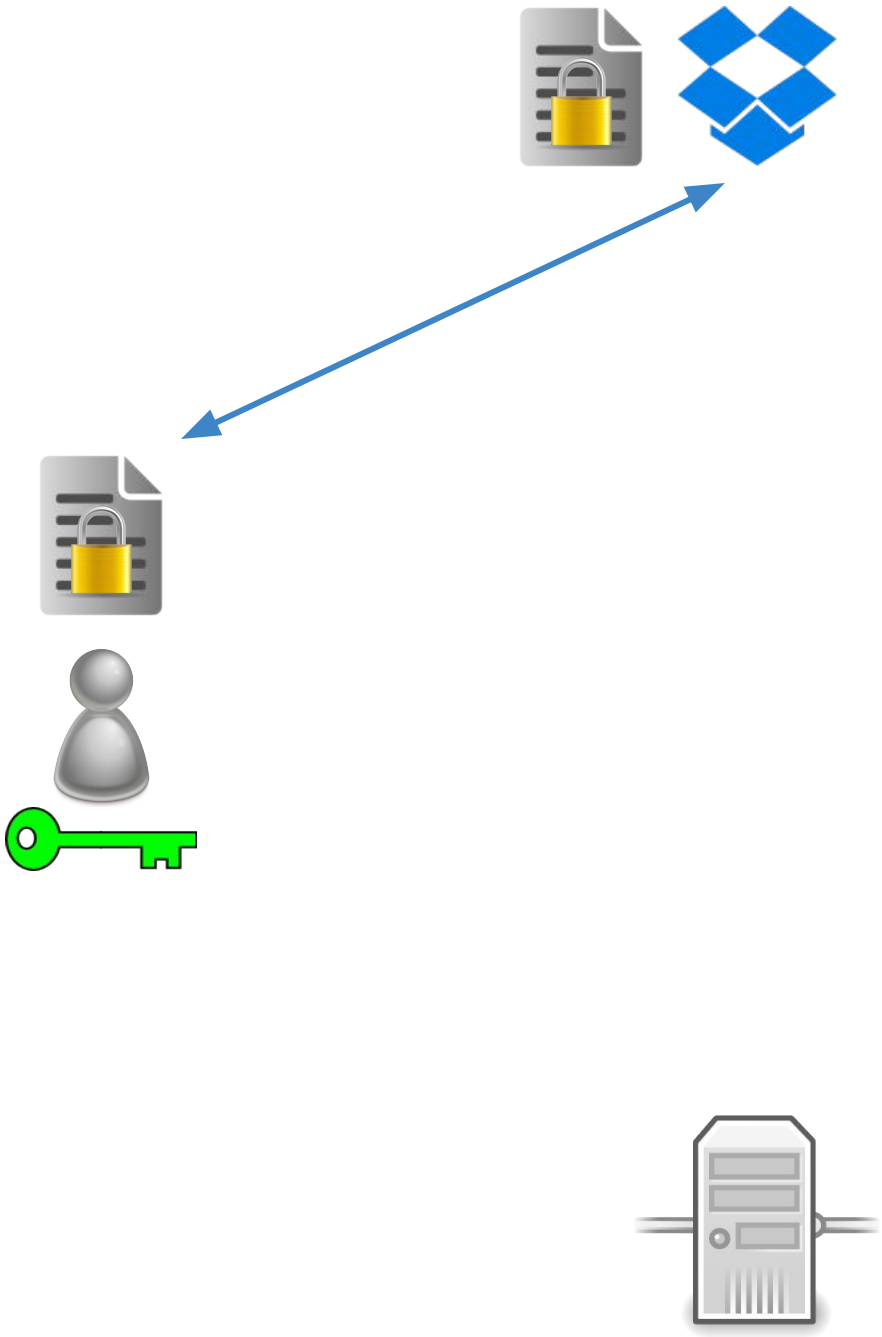
# Fusebox:
# Tutamen-backed Dropbox Client

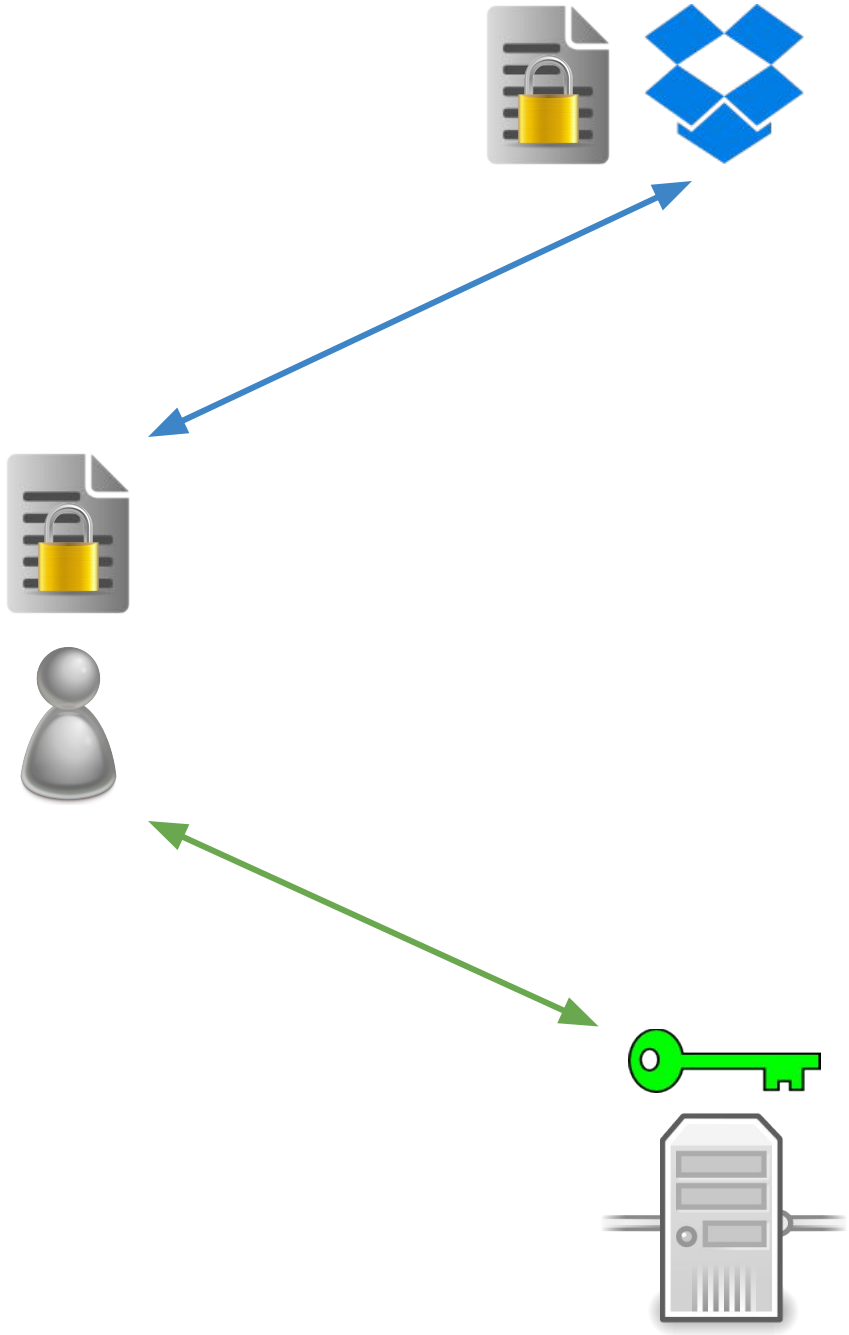
Implementation by Taylor Andrews

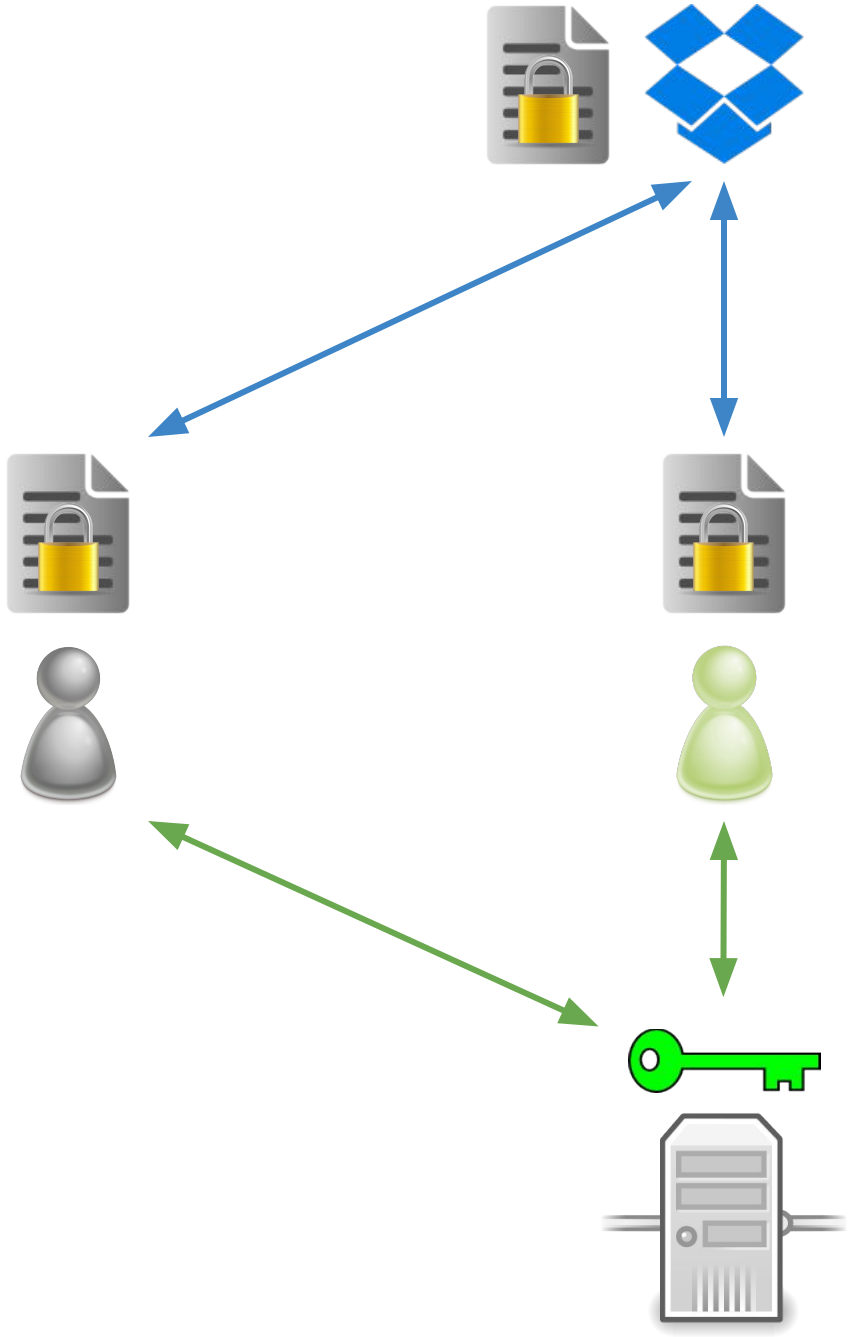


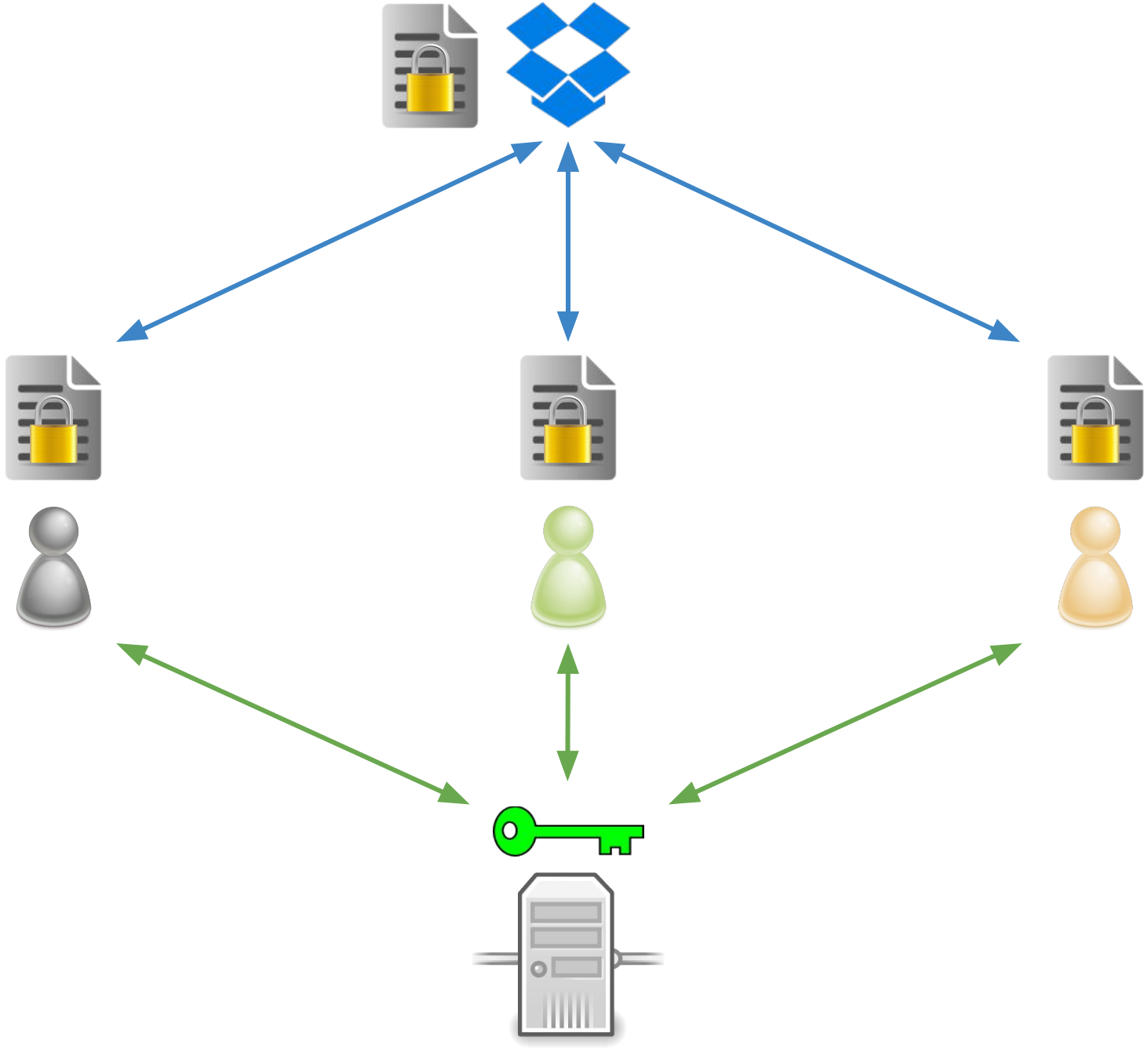


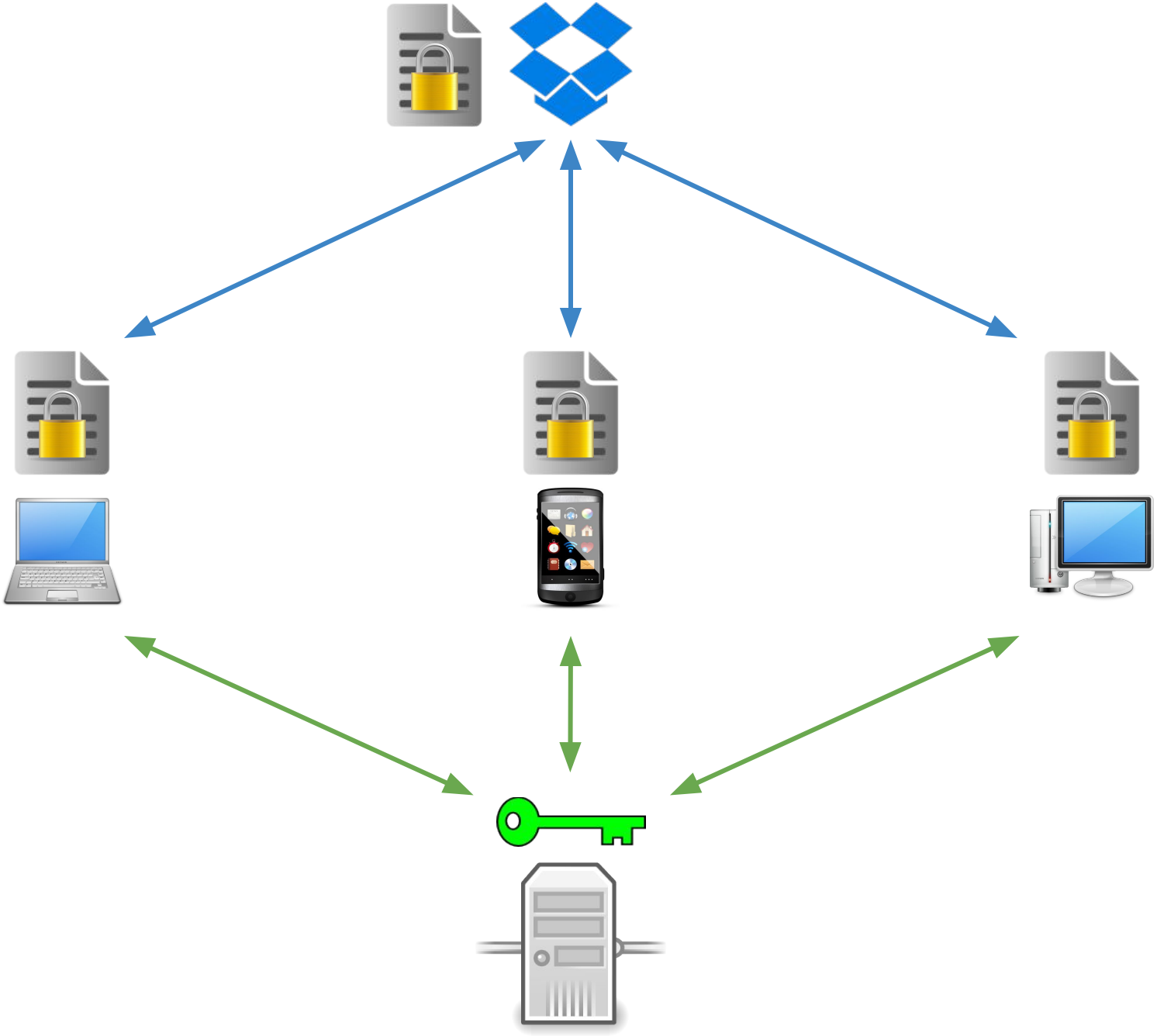






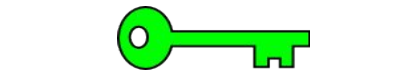


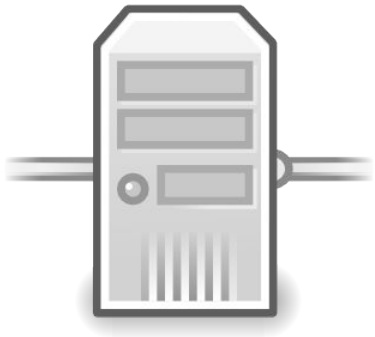


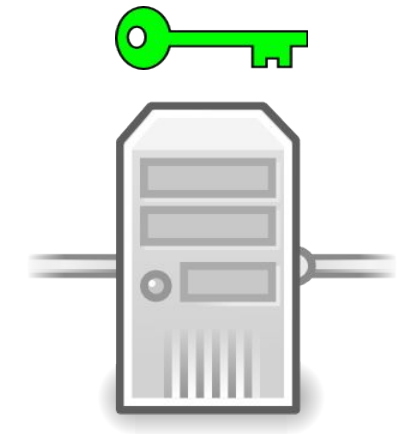


# Tutamen-backed dm-crypt/LUKS FDE

Implementation by Matt Monaco



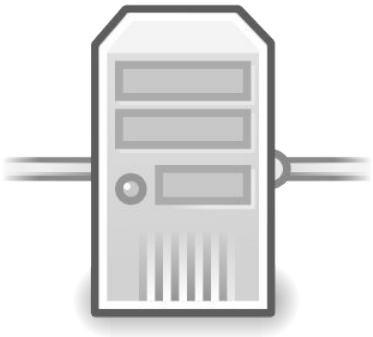




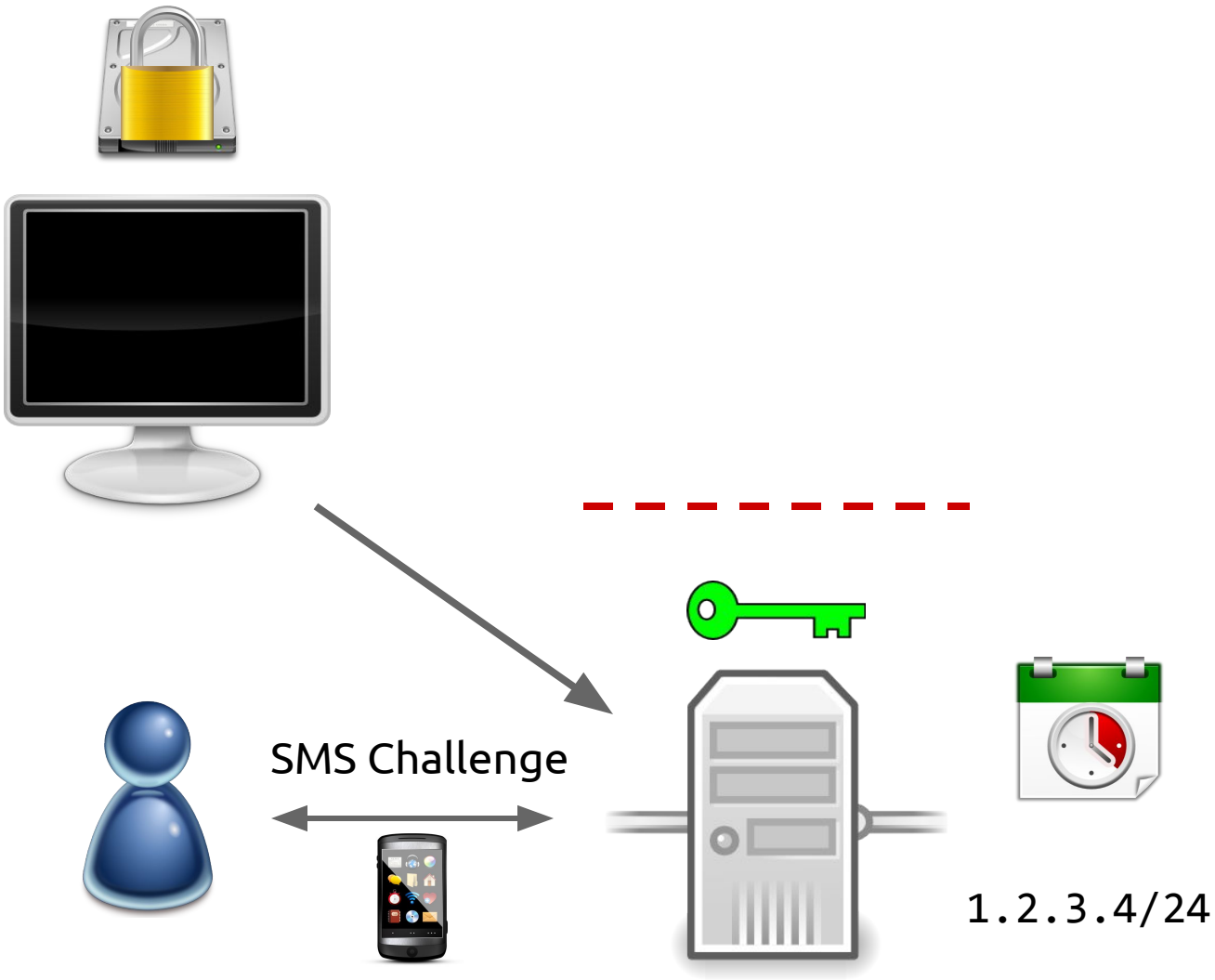
1.2.3.4/24

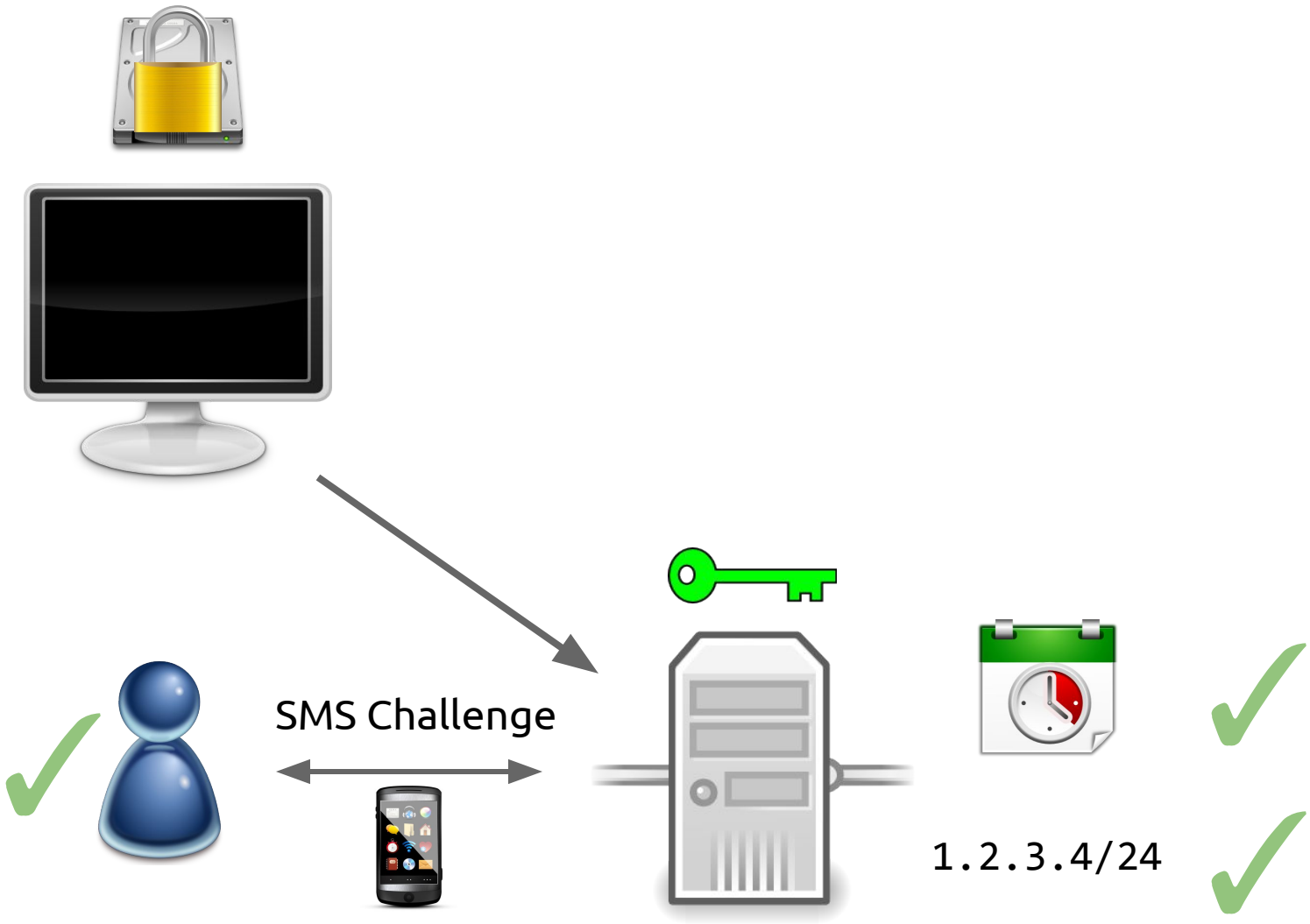

SMS Challenge

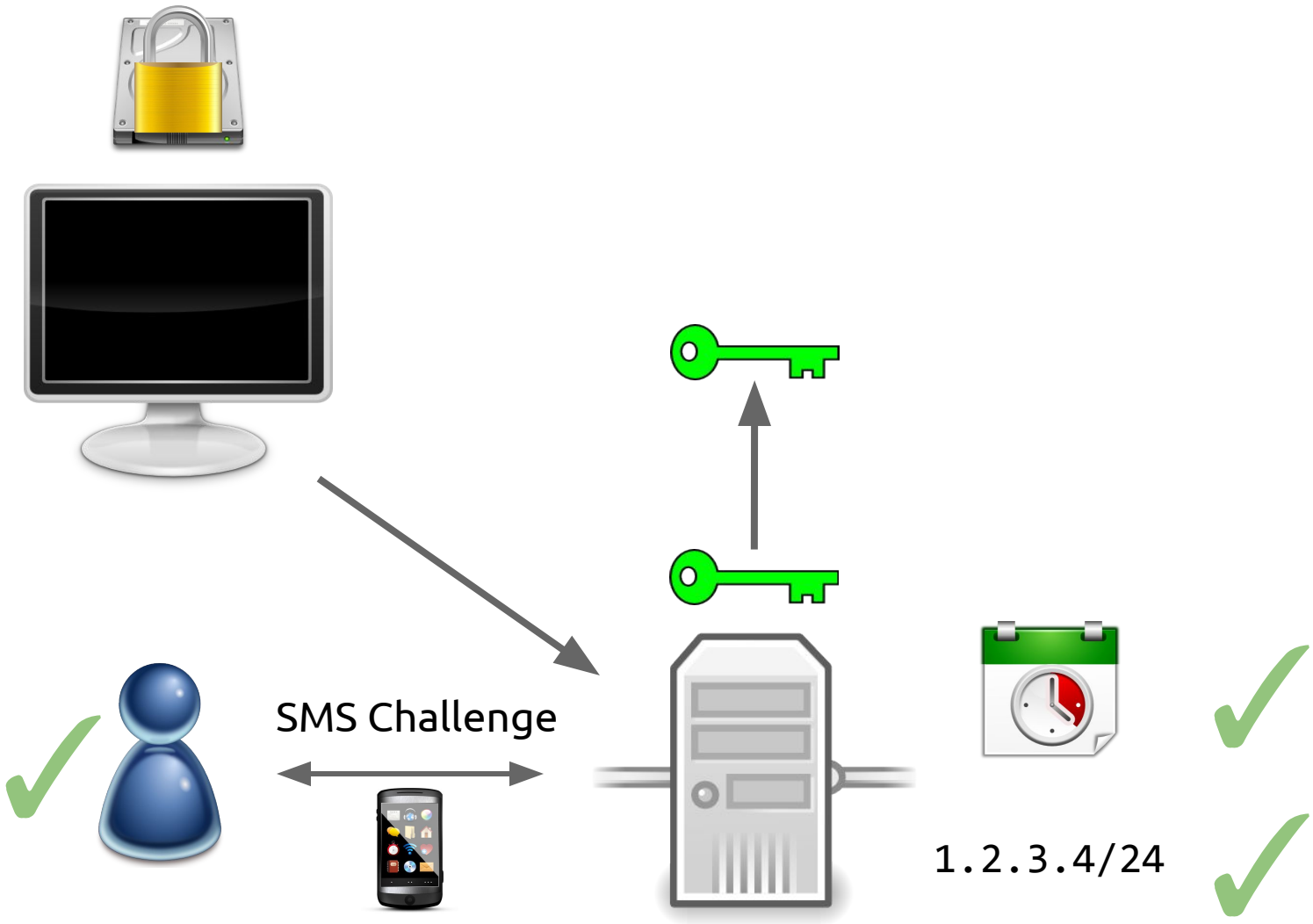
SMS Challenge

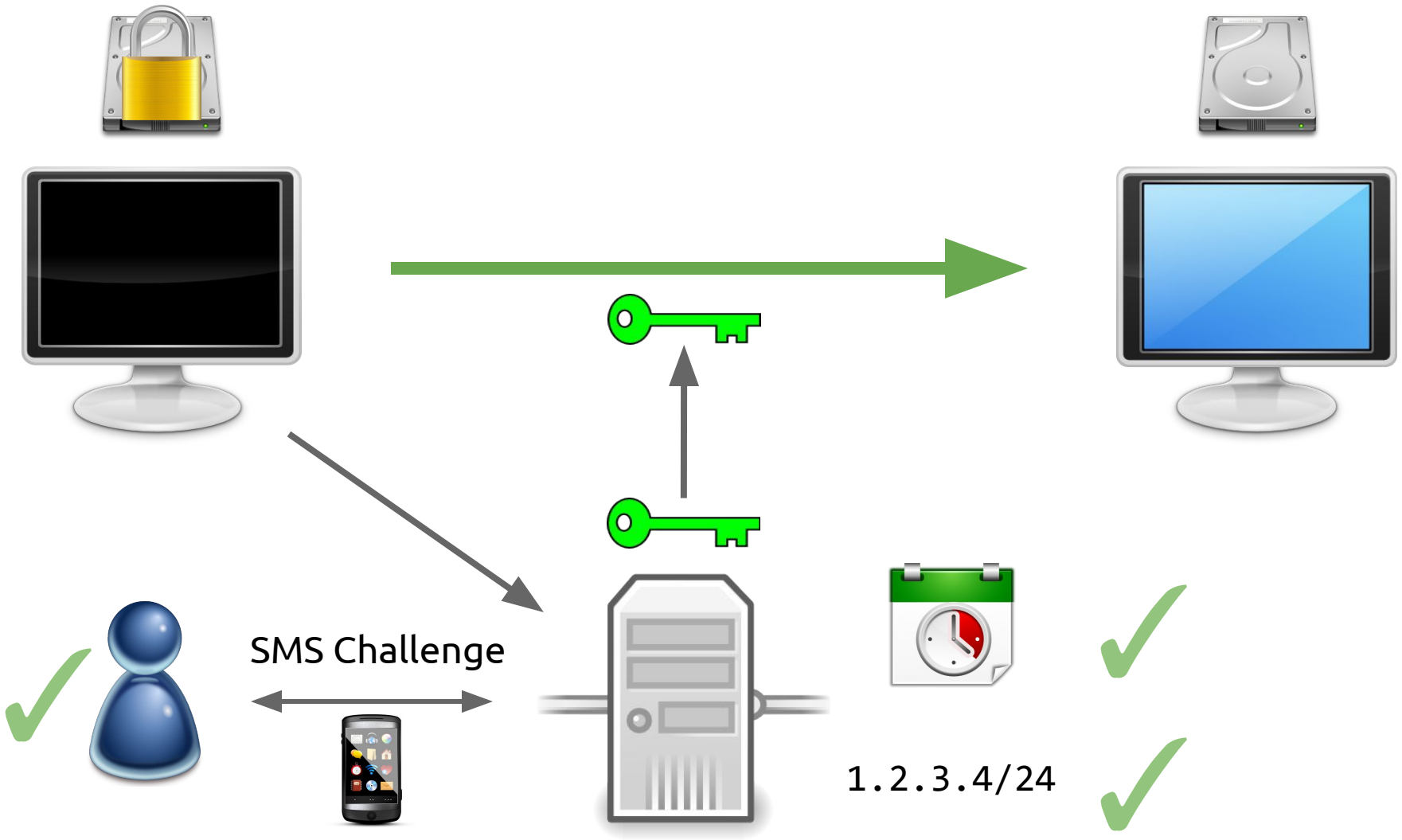


1.2.3.4/24


SMS Challenge
1.2.3.4/24

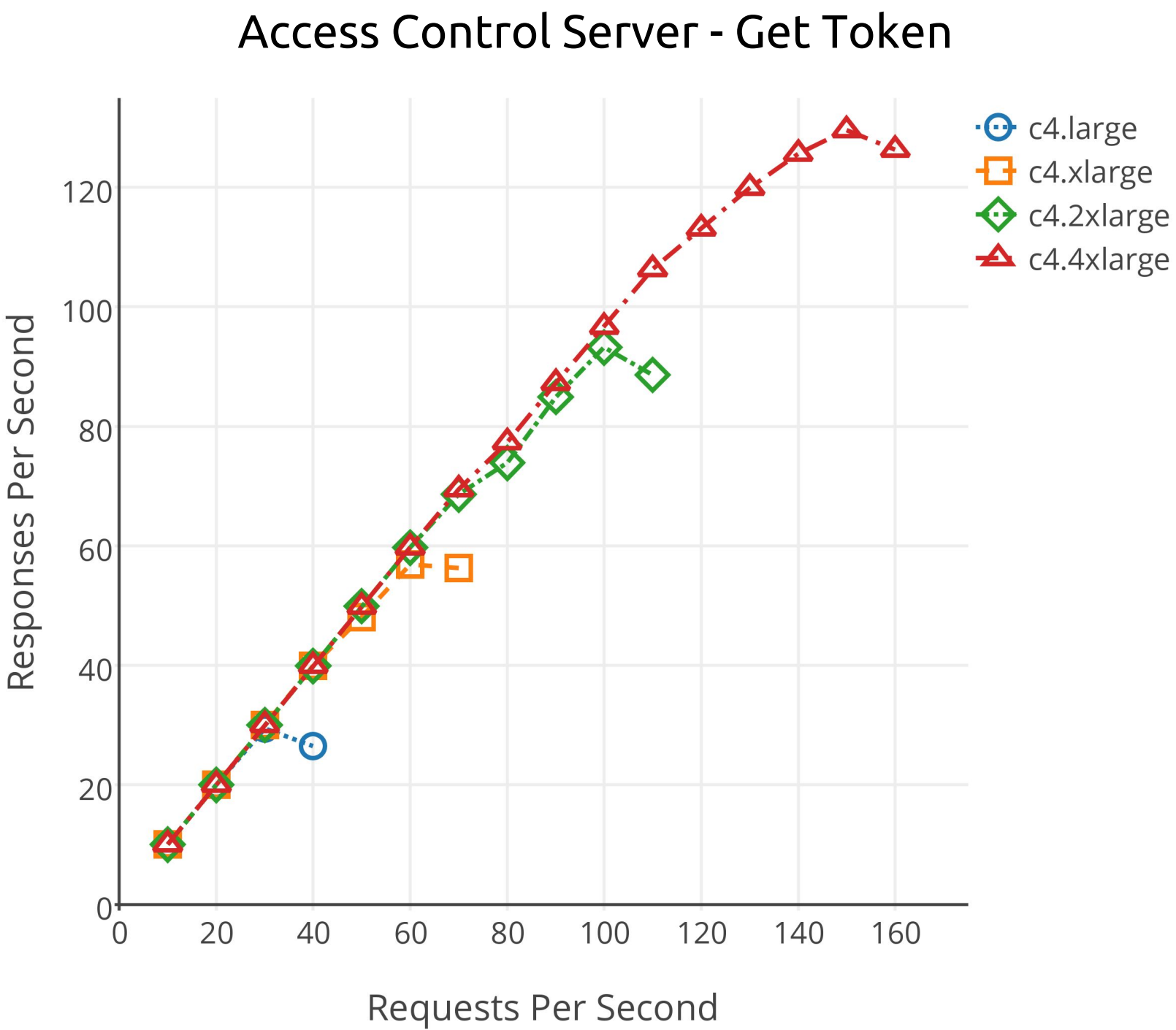

SMS Challenge
1.2.3.4/24

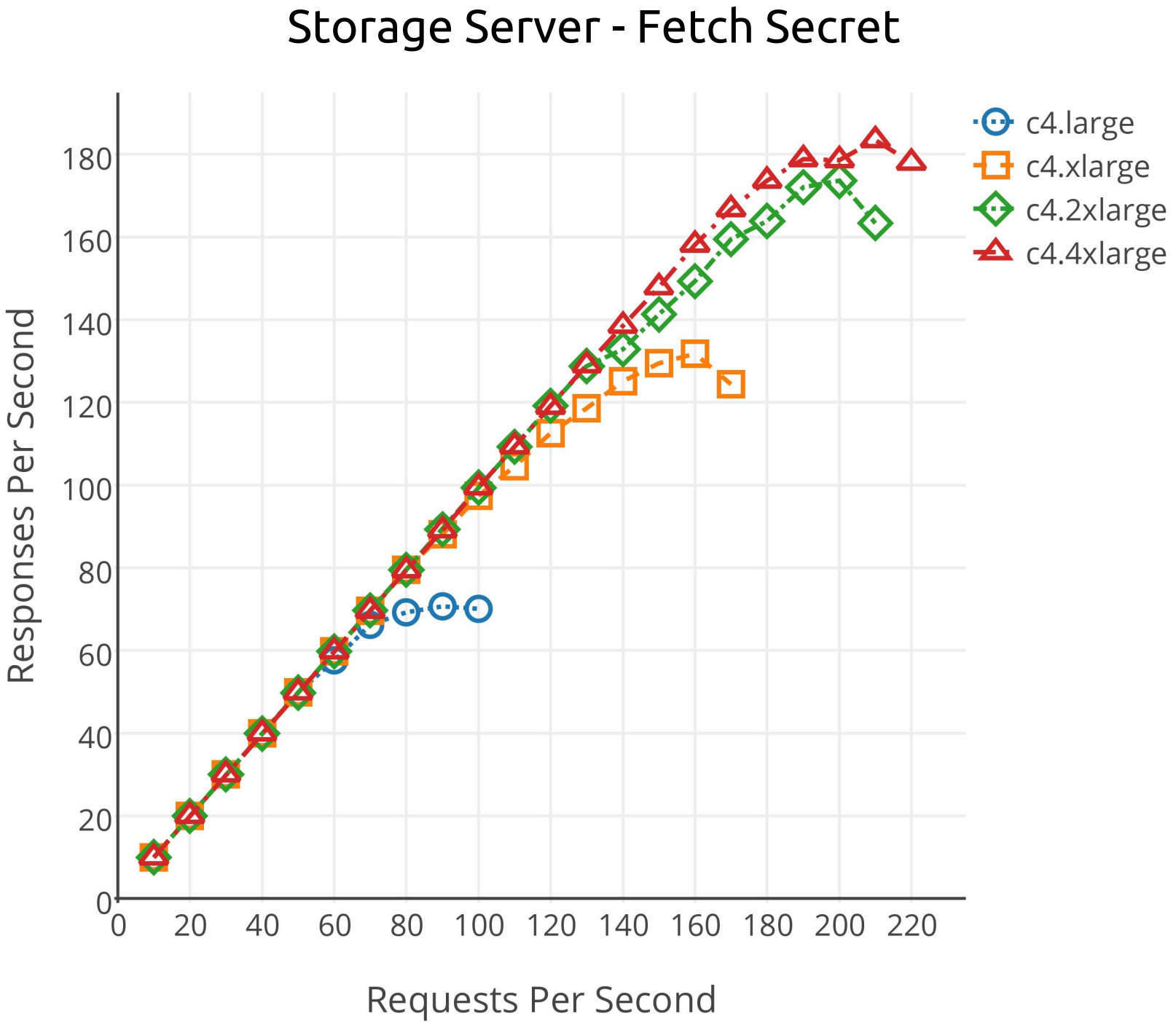

SMS Challenge
1.2.3.4/24


SMS Challenge
1.2.3.4/24

# Evaluation

# Useful Across a Range of Applications


Access Control Server - Get Token
Responses Per Second
Requests Per Second
0
20
40
60
80
100
120
140
160
c4.large
c4.xlarge
c4.2xlarge
c4.4xlarge


Storage Server - Fetch Secret
Responses Per Second
Requests Per Second
0
20
40
60
80
100
120
140
160
180
200
220
c4.large
c4.xlarge
c4.2xlarge
c4.4xlarge

# Conclusion

# Next-Generation Secret Storage as a Service

# Next-Generation Secret Storage as a Service

# Tutamen

## Flexible Authentication

Plugins for Multi-factor, Out-of-Band, Etc Auth

## Minimally Trusted Infrastructure

Sharding Across Multiple Servers

## Beyond a Single Administrative Domain

Distributed Federation Between Servers

Thank You

Questions?

# Extra Slides

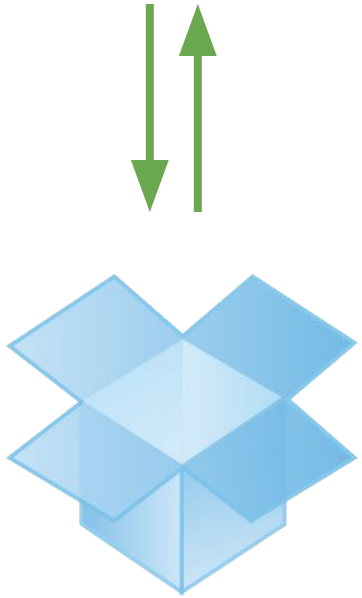
How can we secure
and control our data?

*(even in the presence third parties)*

*(while also supporting modern use cases)*

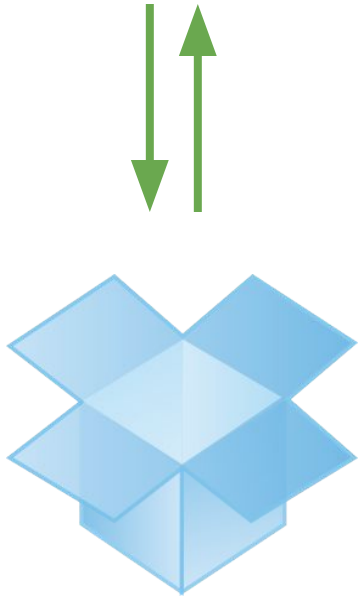
# Client-Side Encryption?

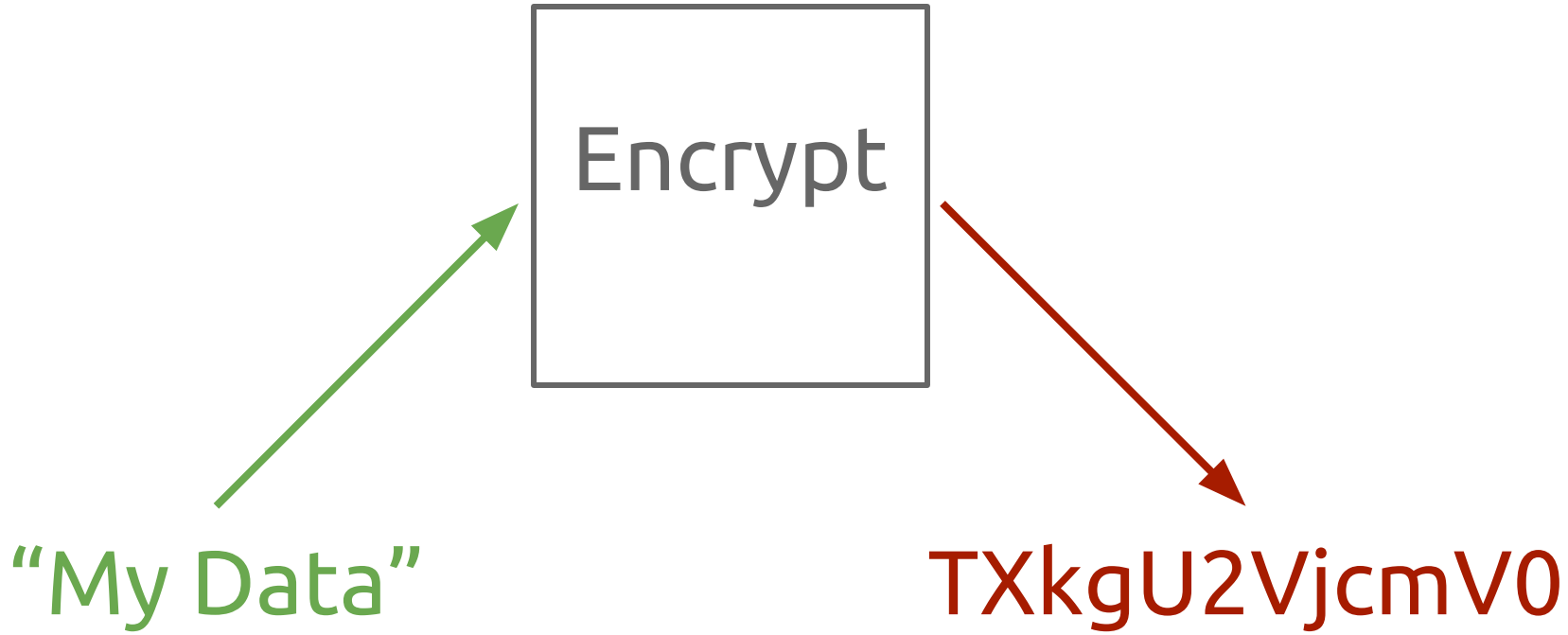
“My Data”

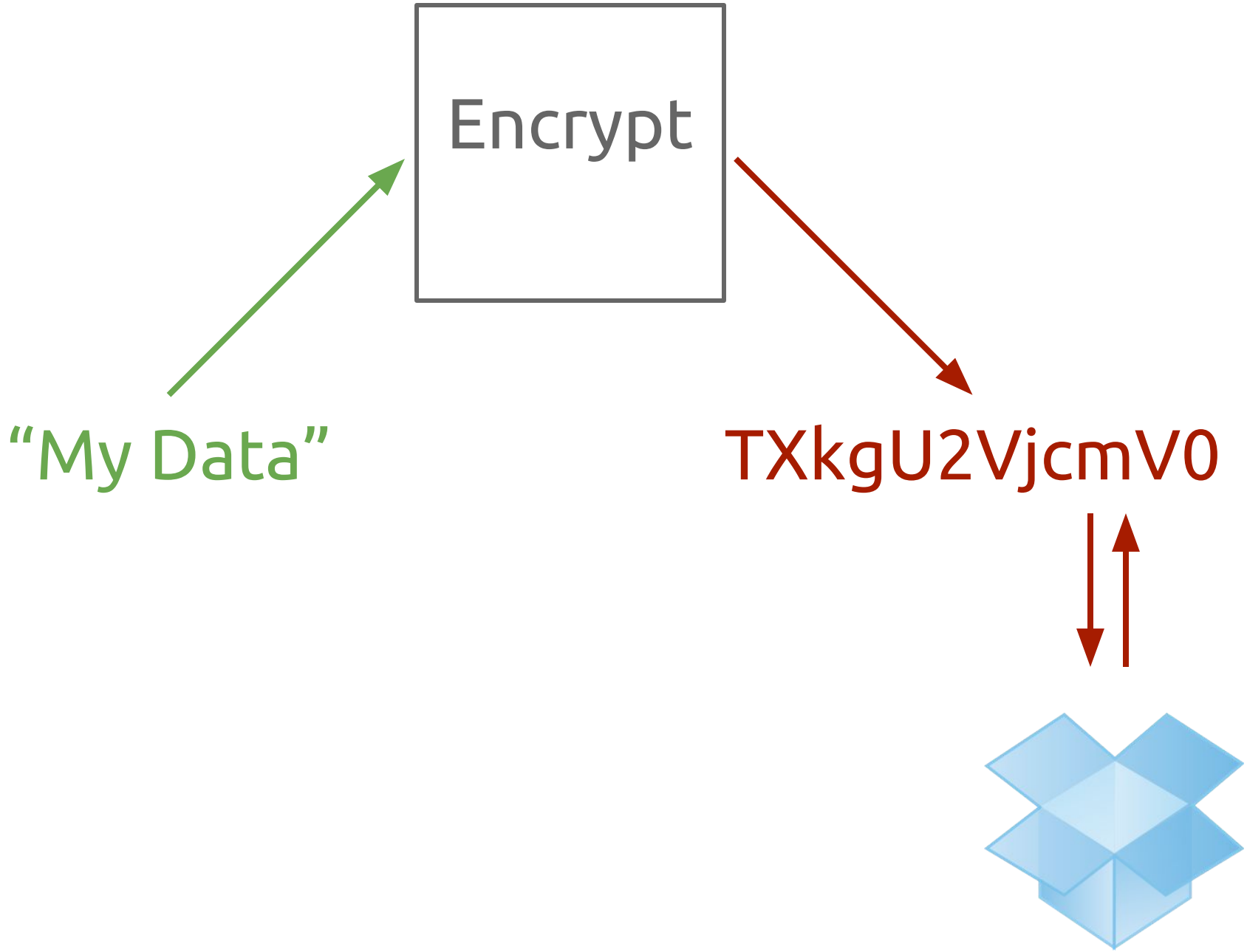


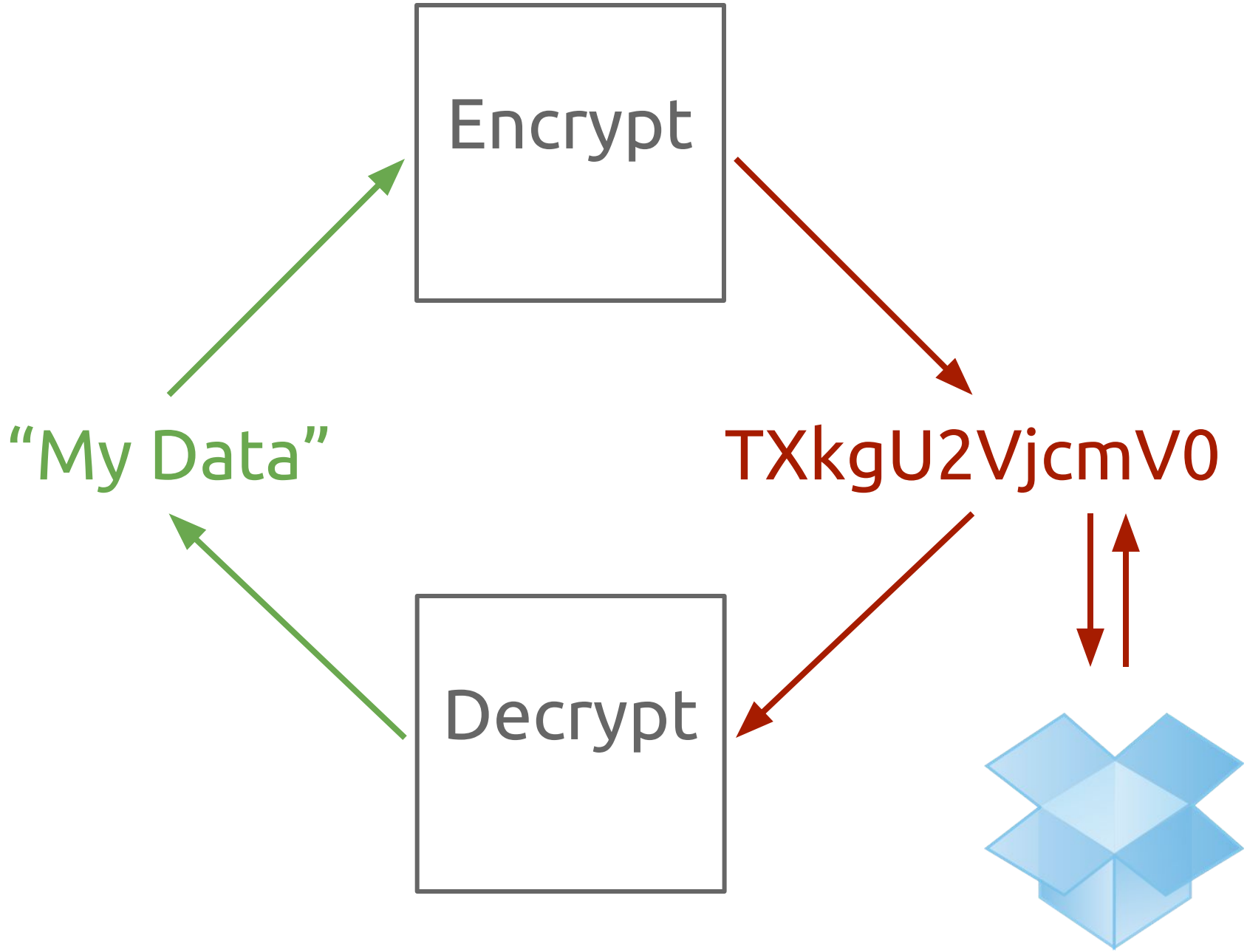
*Cryptography!*

“My Data”




Encrypt
“My Data”
TXkgU2VjcmV0

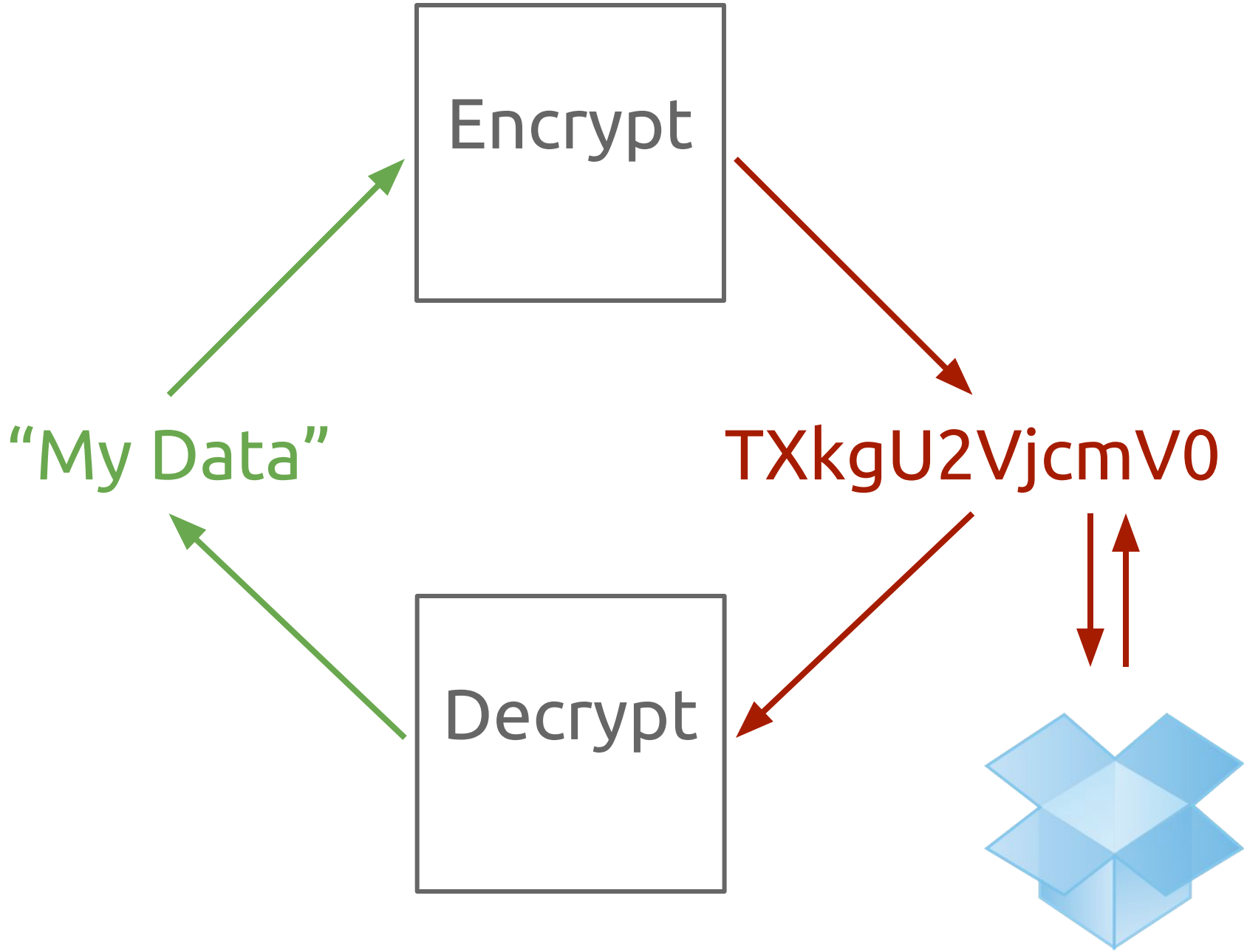

Encrypt
“My Data”
TXkgU2VjcmV0

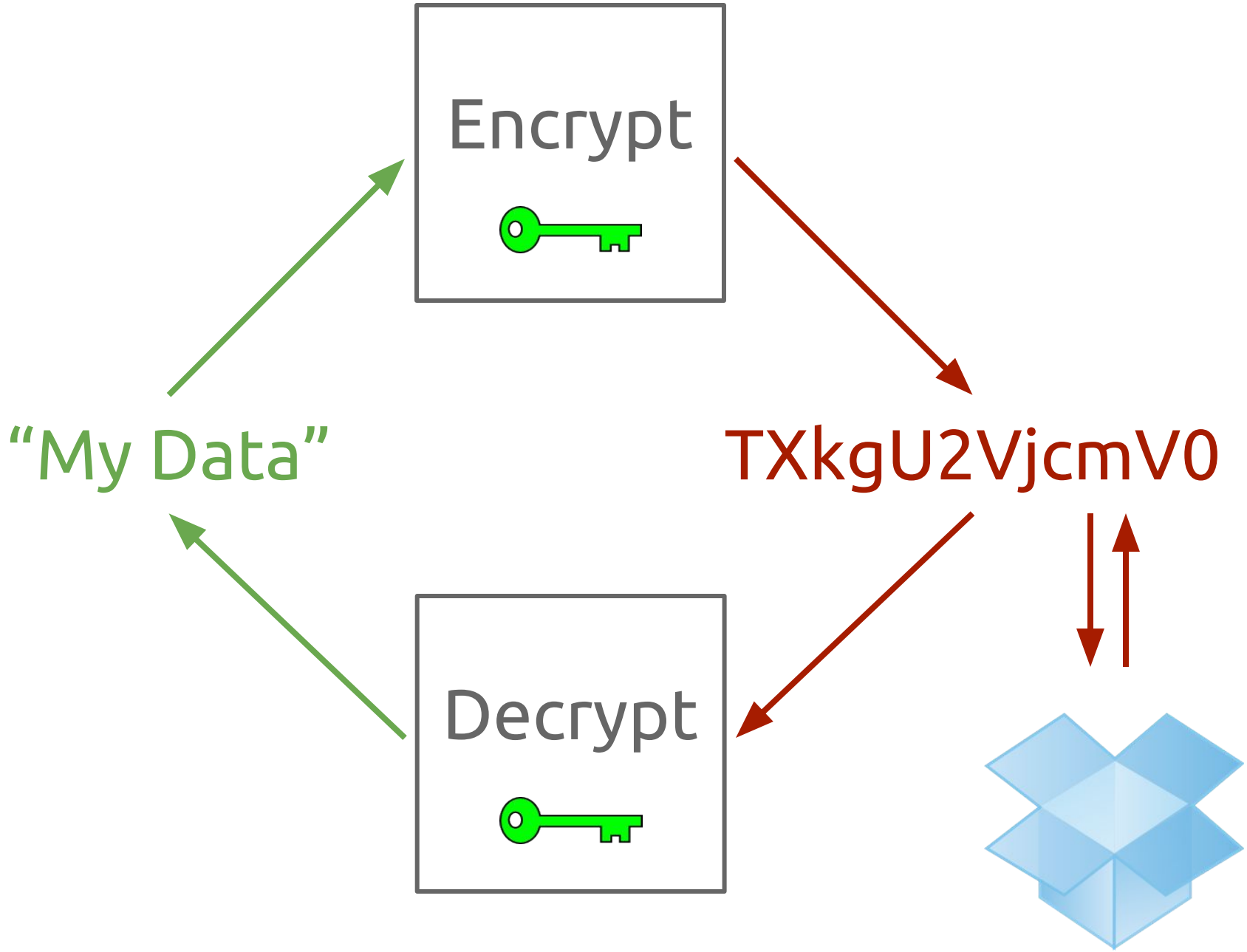

Encrypt
“My Data”
TXkgU2VjcmV0
Decrypt

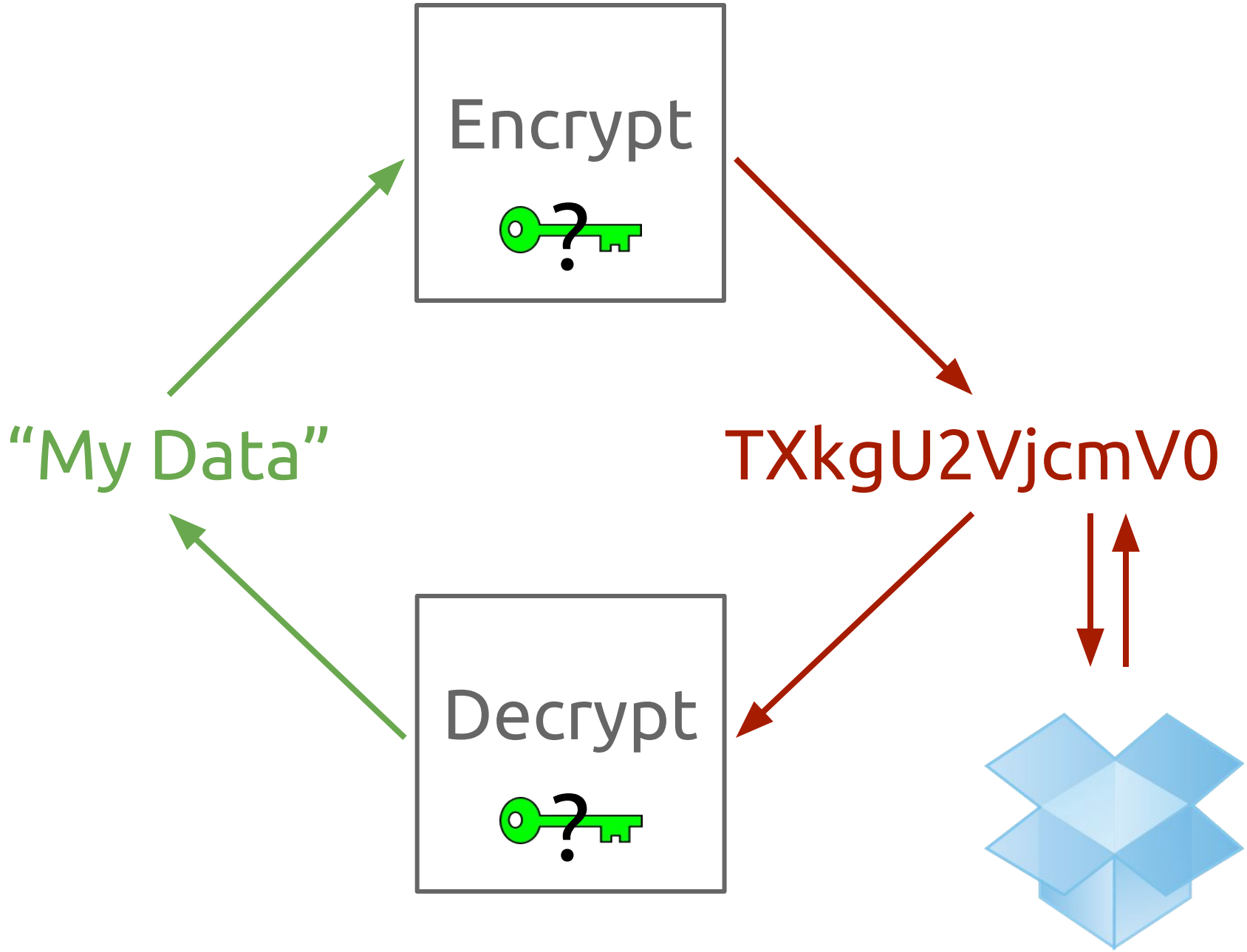
*Cryptography!*

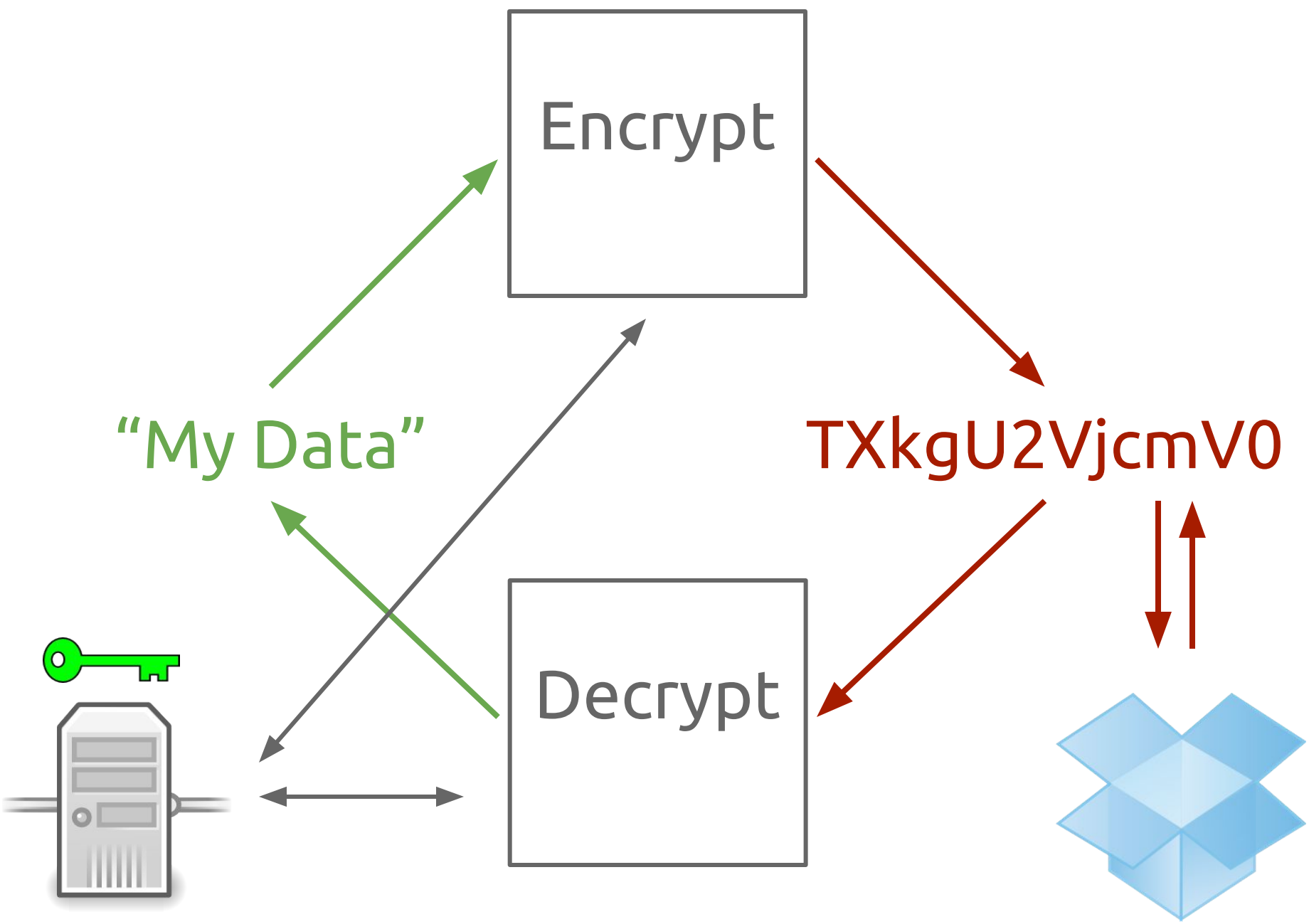
+ - * / ^

*Cryptography!*

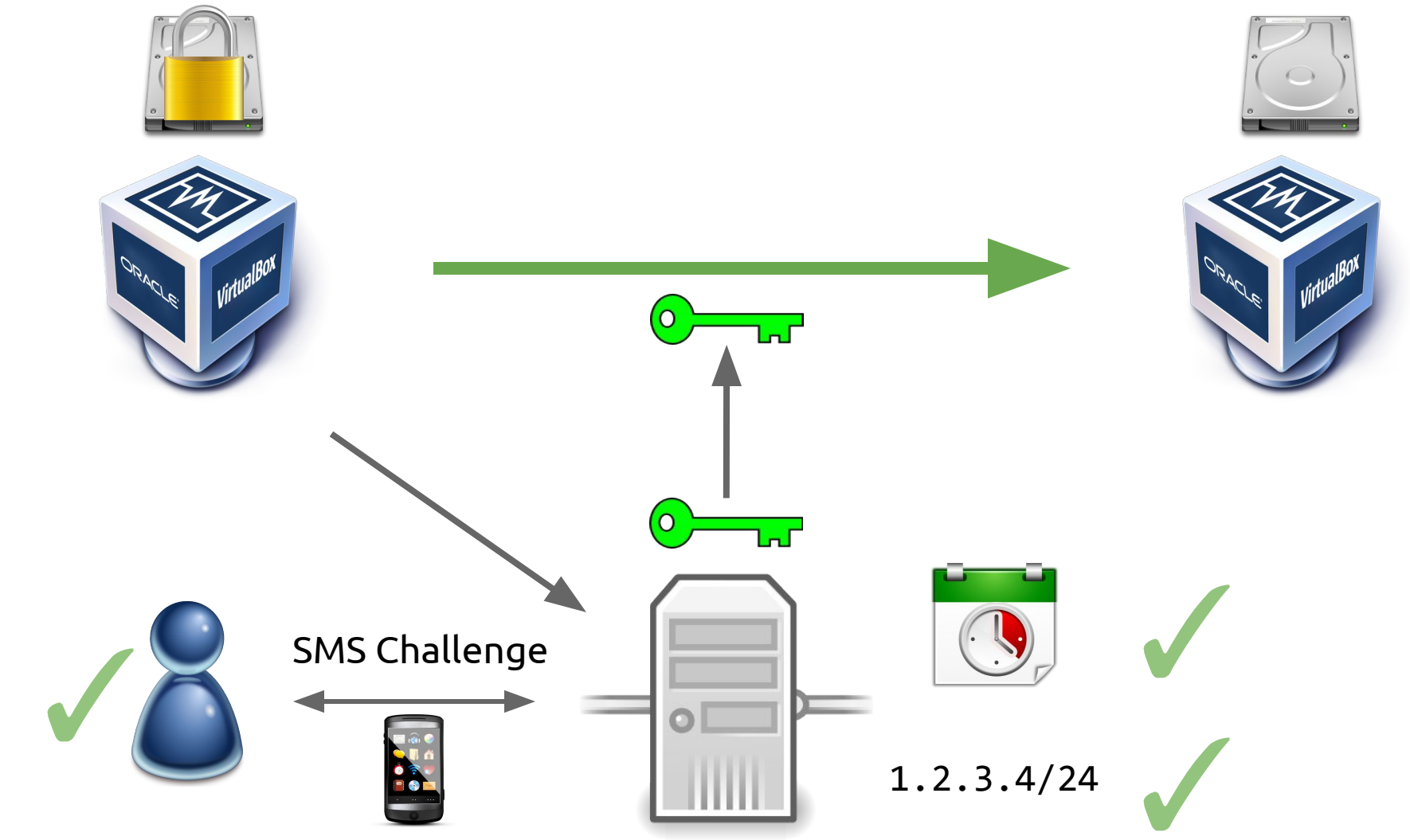

Encrypt
“My Data”
TXkgU2VjcmV0
Decrypt


Encrypt
"My Data"
TXkgU2VjcmV0
Decrypt

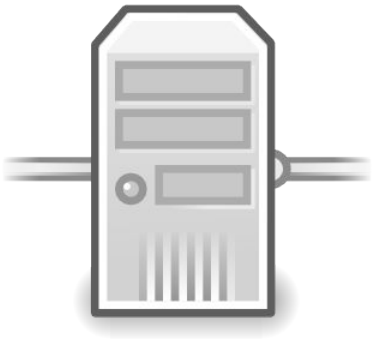

Encrypt
“My Data”
TXkgU2VjcmV0
Decrypt

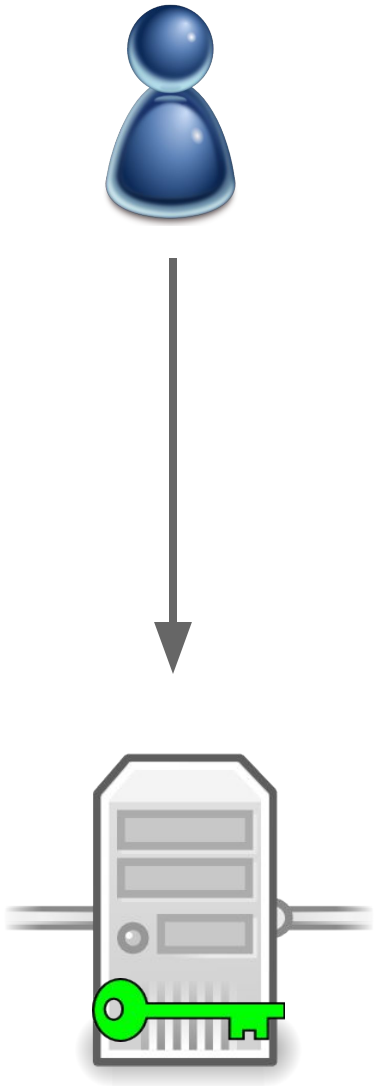

Encrypt
“My Data”
TXkgU2VjcmV0
Decrypt

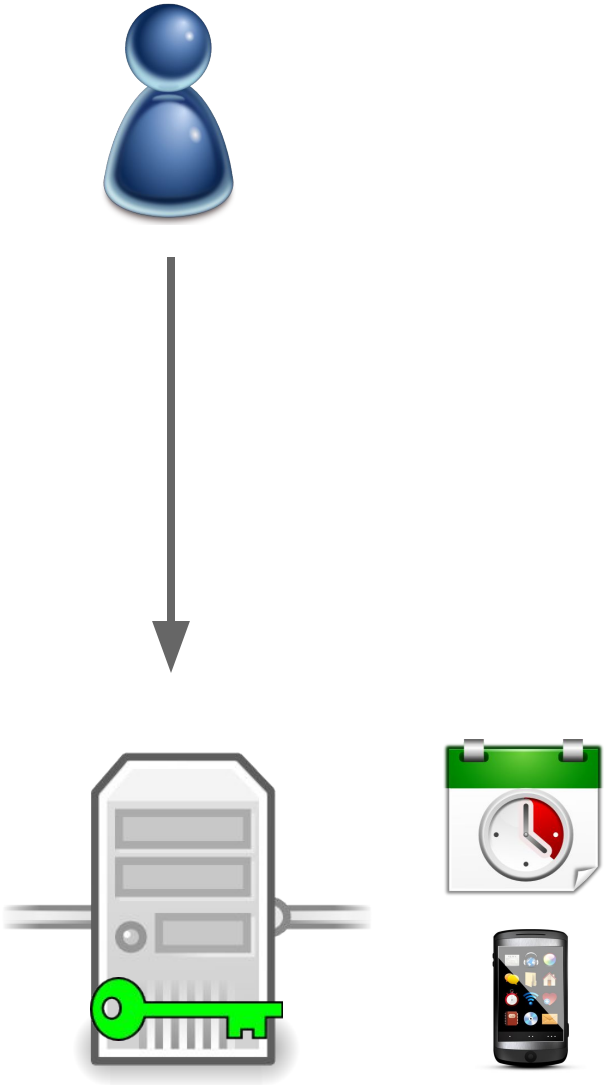
# Tutamen-backed QEMU VM Encryption

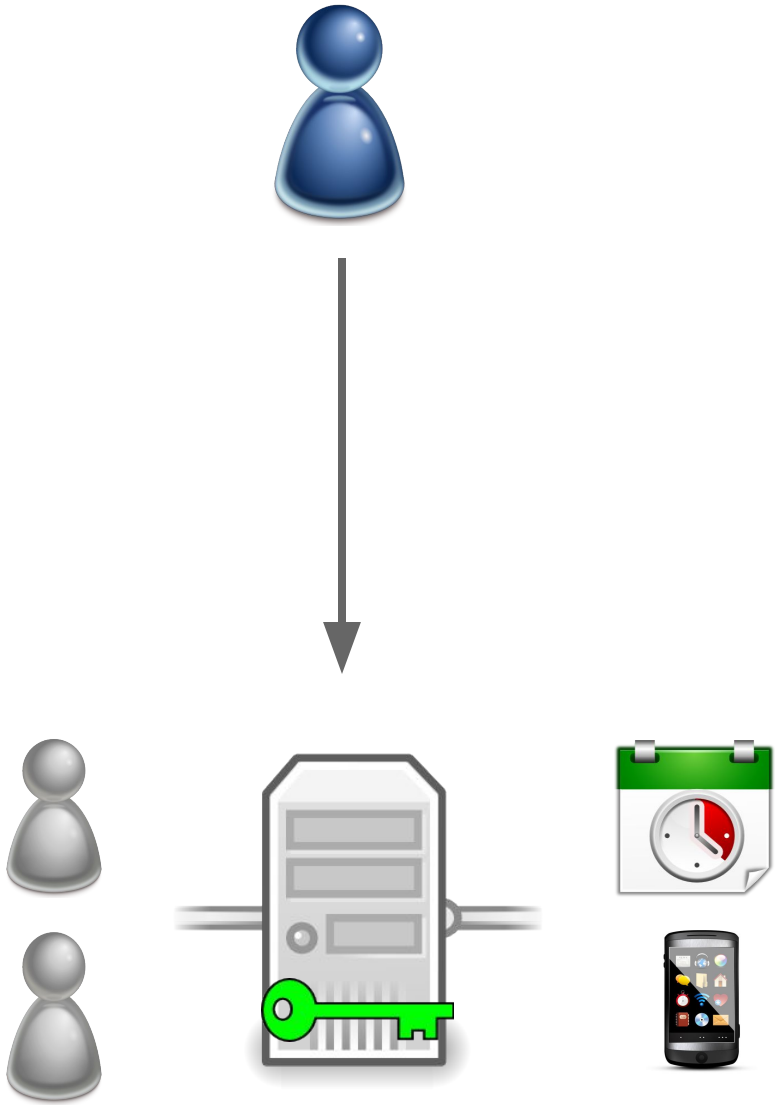

SMS Challenge
1.2.3.4/24

# Tutamen Management Utility

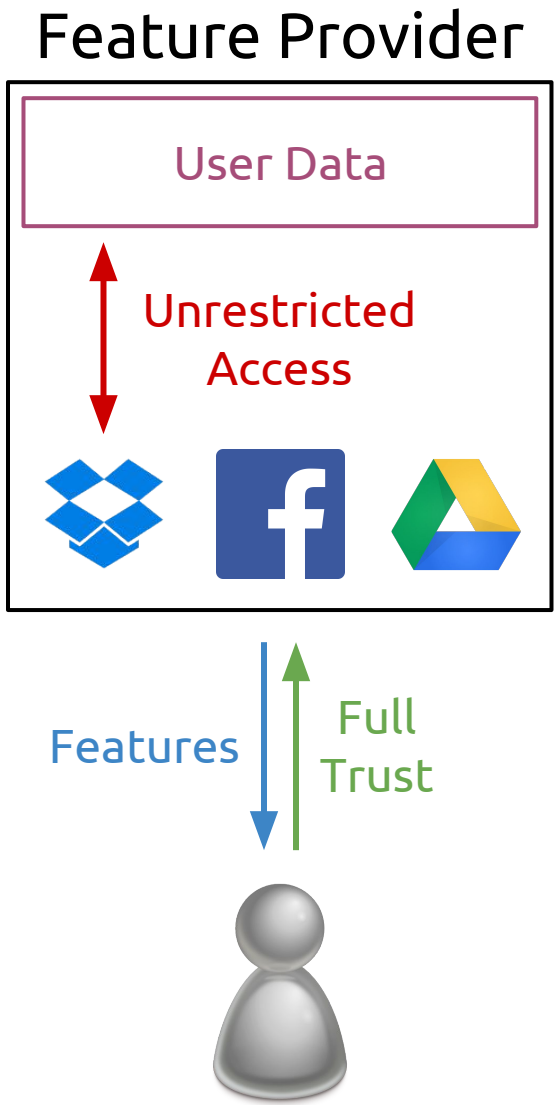




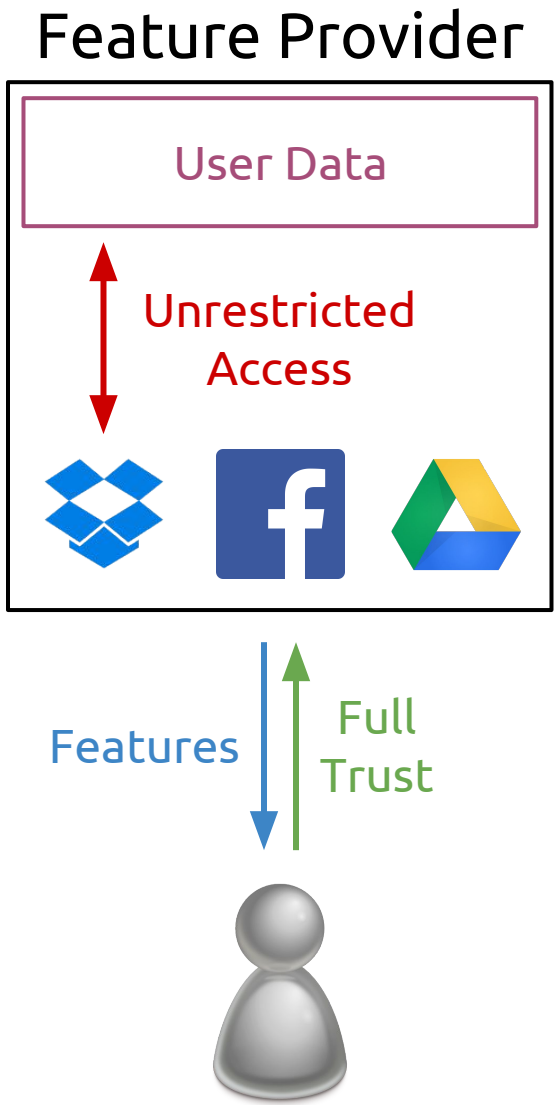




# Traditional Trust Model



# Traditional Trust Model



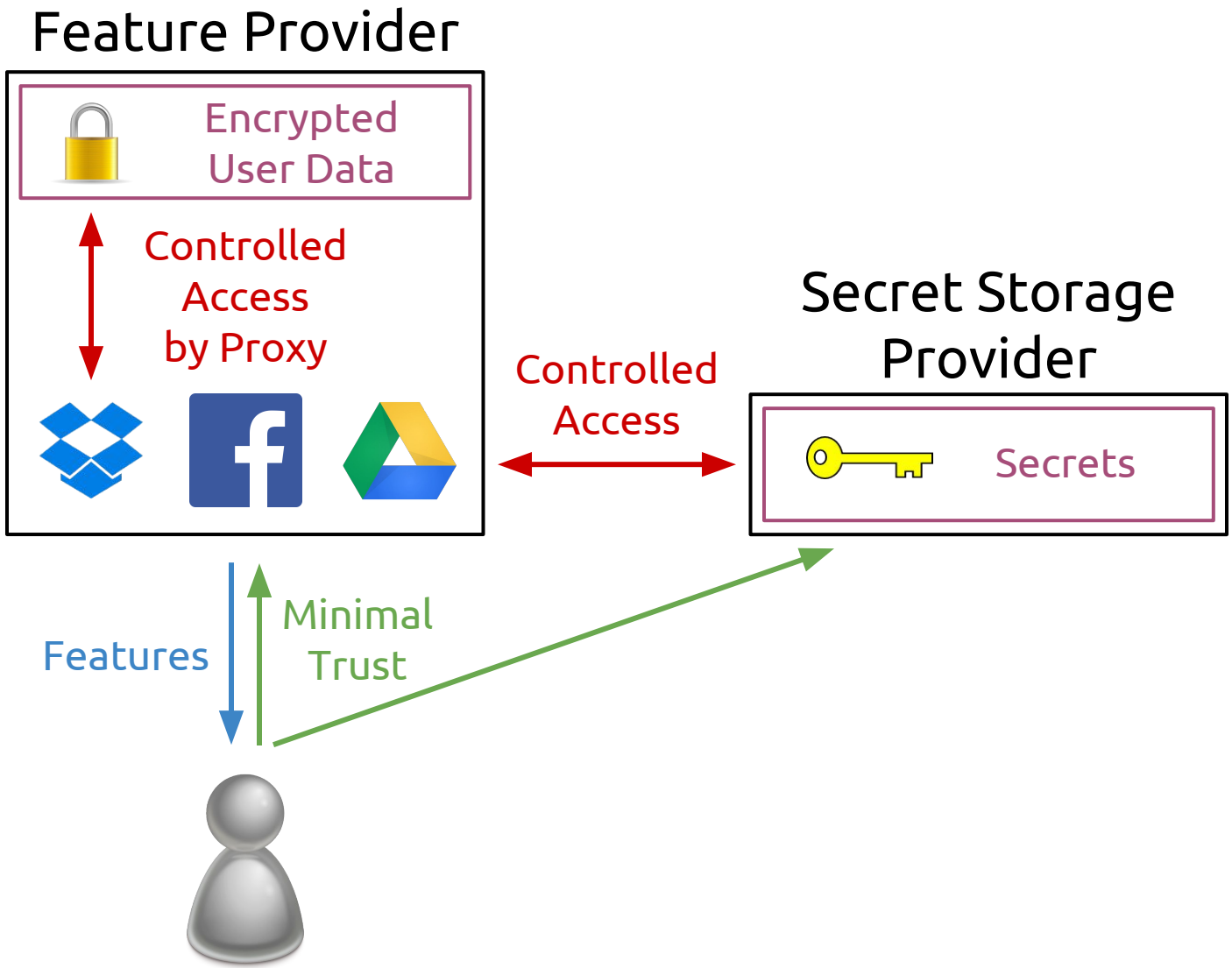

Storage (S)

Access (R)

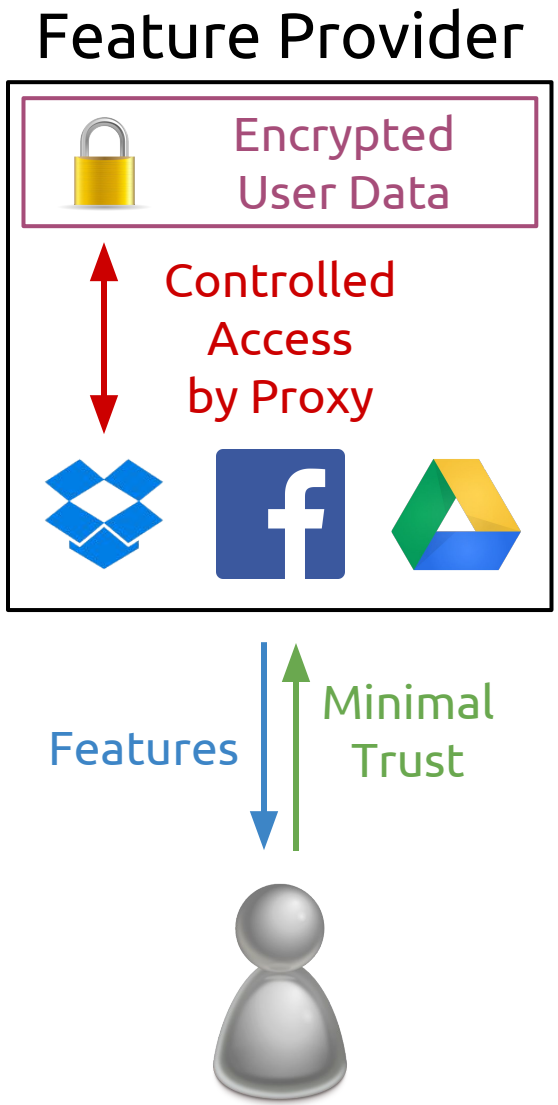
Manipulation (W)

Meta-Analysis (M)

# SSaaS Trust Model



# SSaaS Trust Model




Storage (S)

~~Access (R)~~

~~Manipulation (W)~~

Meta-Analysis (M)

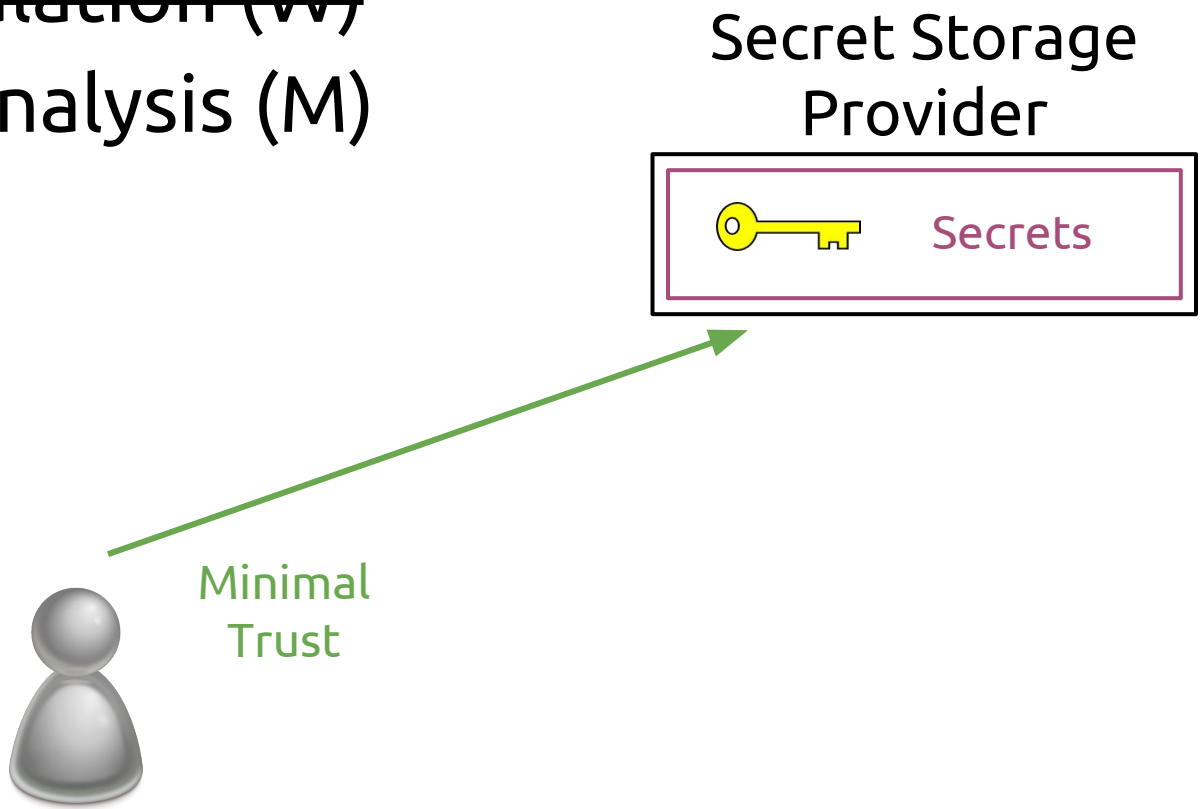
# SSaaS Trust Model

Storage (S)

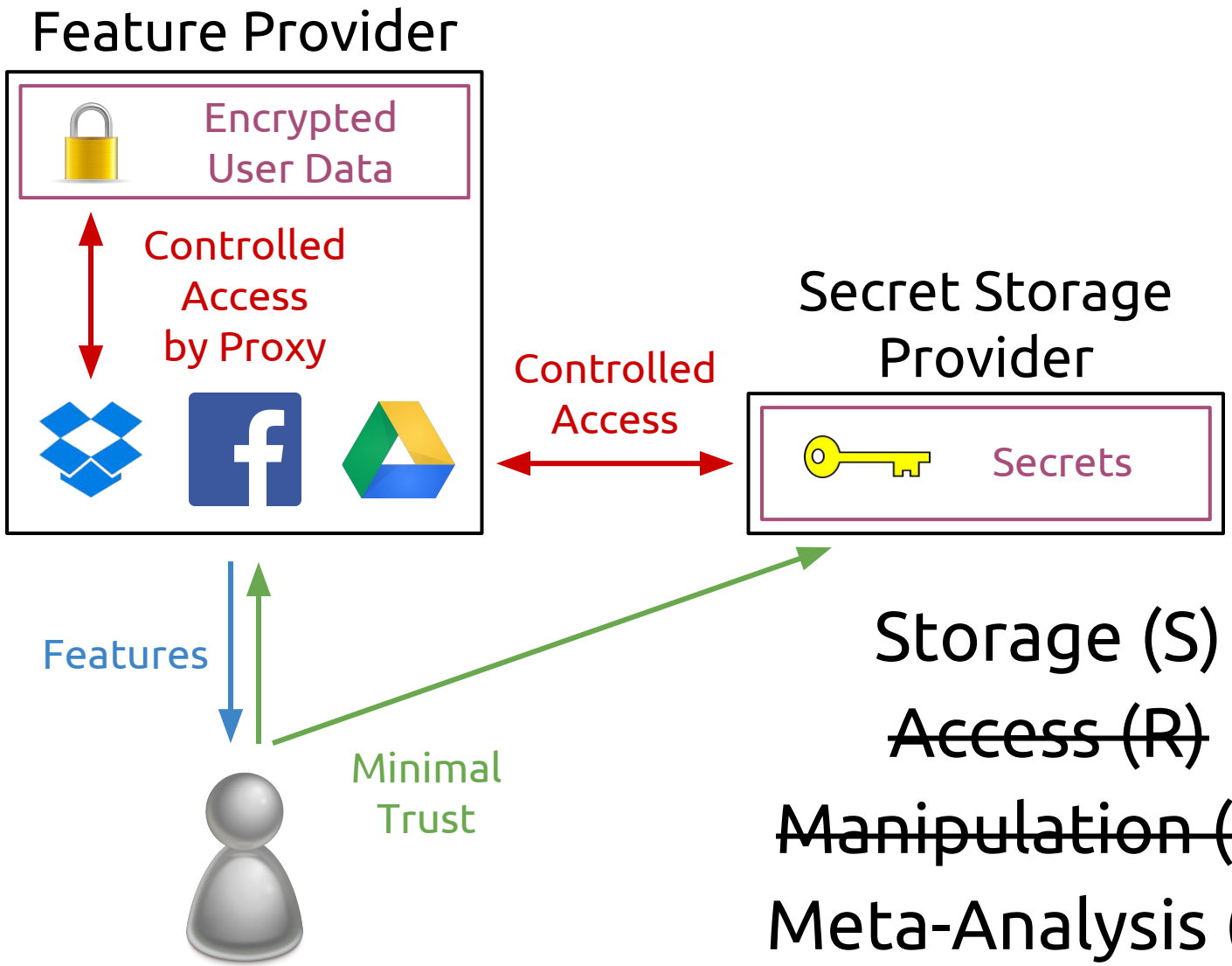
~~Access (R)~~

~~Manipulation (W)~~

Meta-Analysis (M)



# SSaaS Trust Model



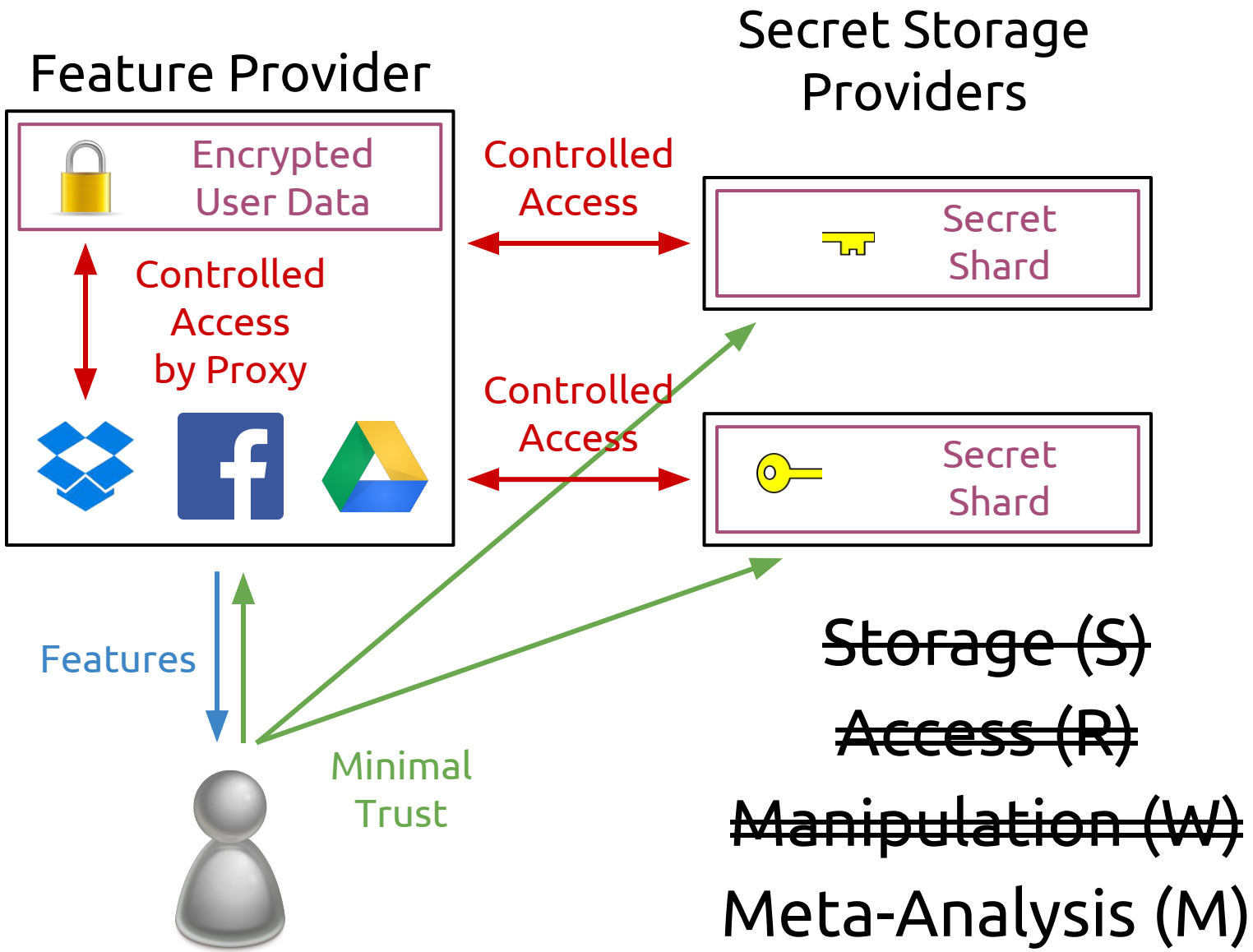

Storage (S)

~~Access (R)~~

~~Manipulation (W)~~

Meta-Analysis (M)

# SSaaS Trust Model




~~Storage (S)~~

~~Access (R)~~

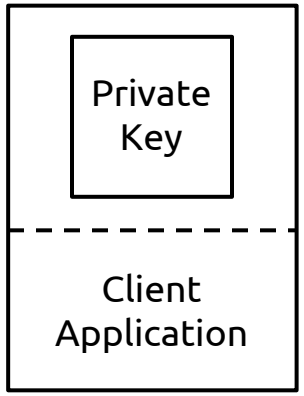
~~Manipulation (W)~~

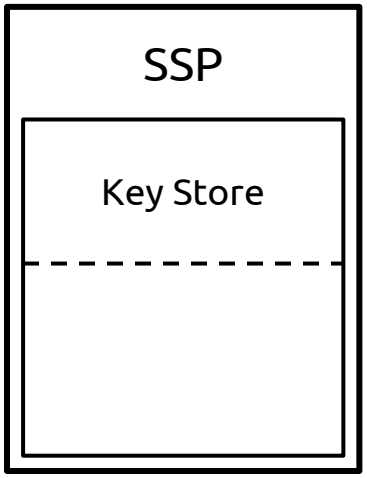
Meta-Analysis (M)

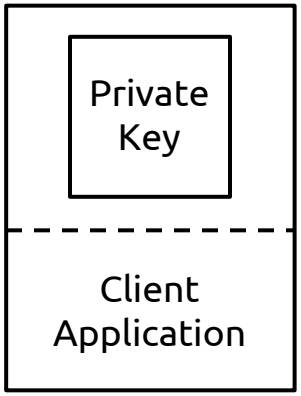
# SSaaS

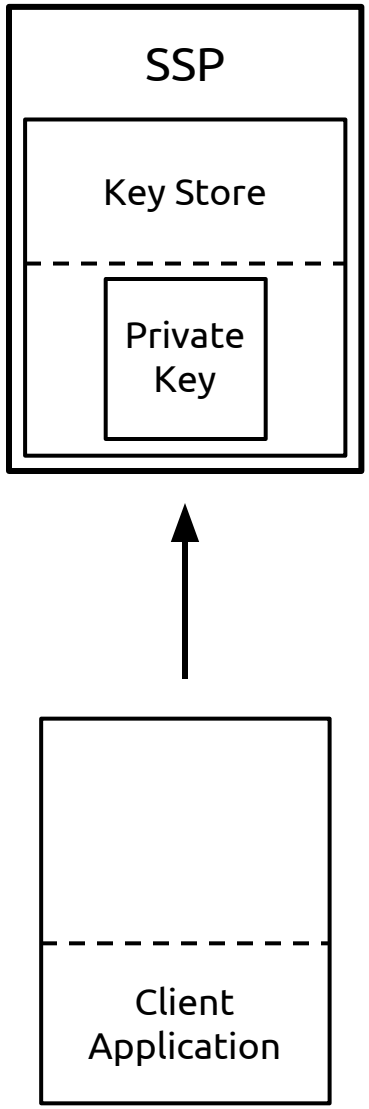
## Security & Trust

# Single SSP


Private
Key
Client
Application


SSP
Key Store


Private
Key
Client
Application


SSP
Key Store
Private
Key
Client
Application

Should we trust

a single provider?

# Maybe

# Incentives aligned with upholding trust

Incentives aligned with upholding trust

Reputation at stake

Incentives aligned with upholding trust

Reputation at stake

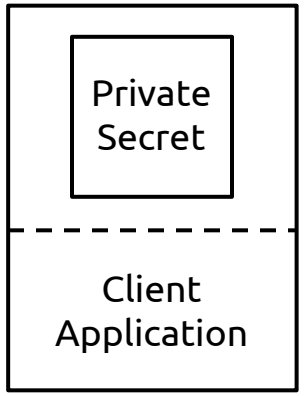
Still a “minimally trusted” entity

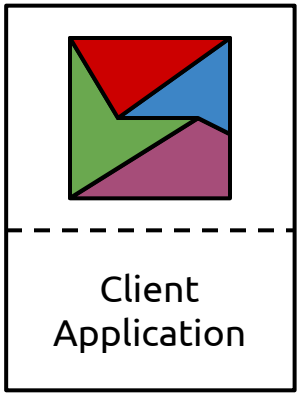
Must we trust

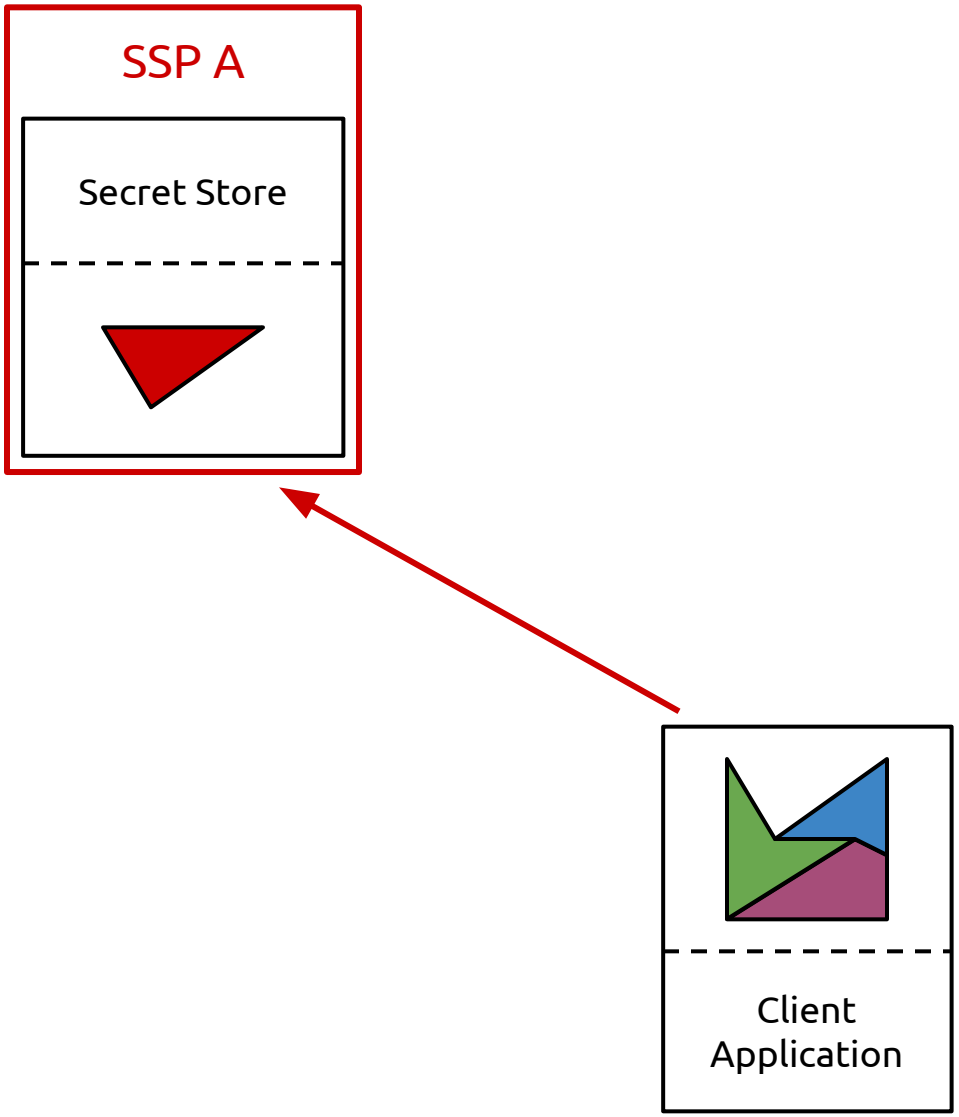
a single provider?

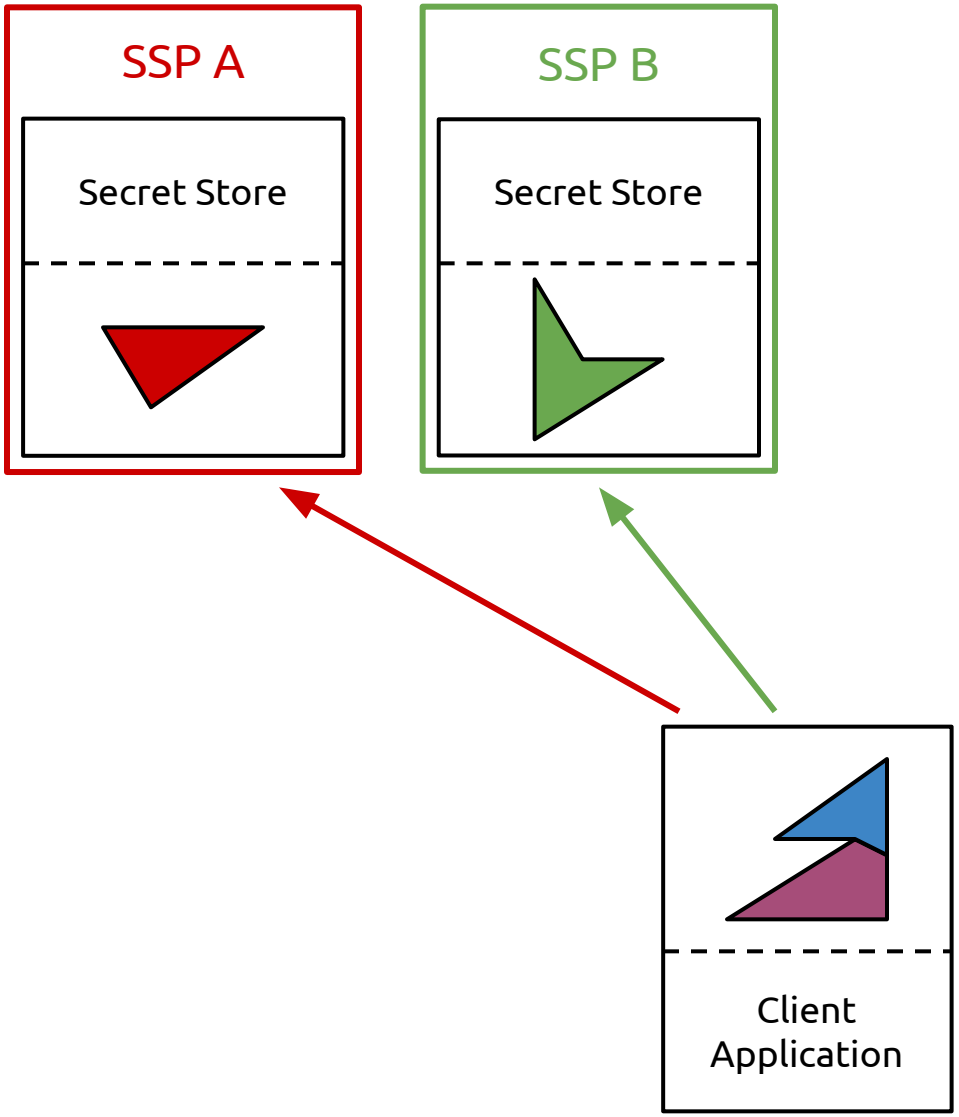
No

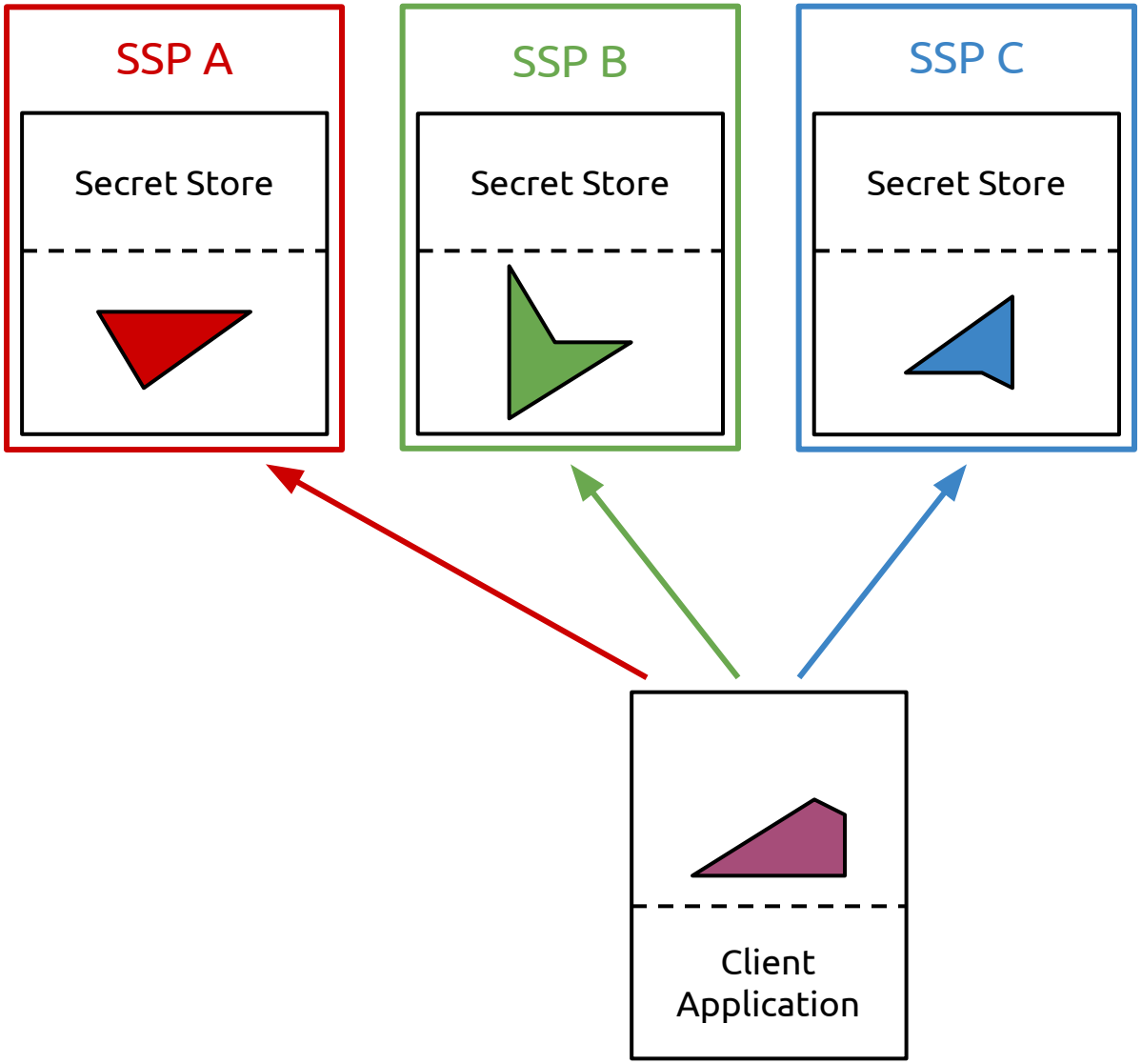
# Multiple SSPs

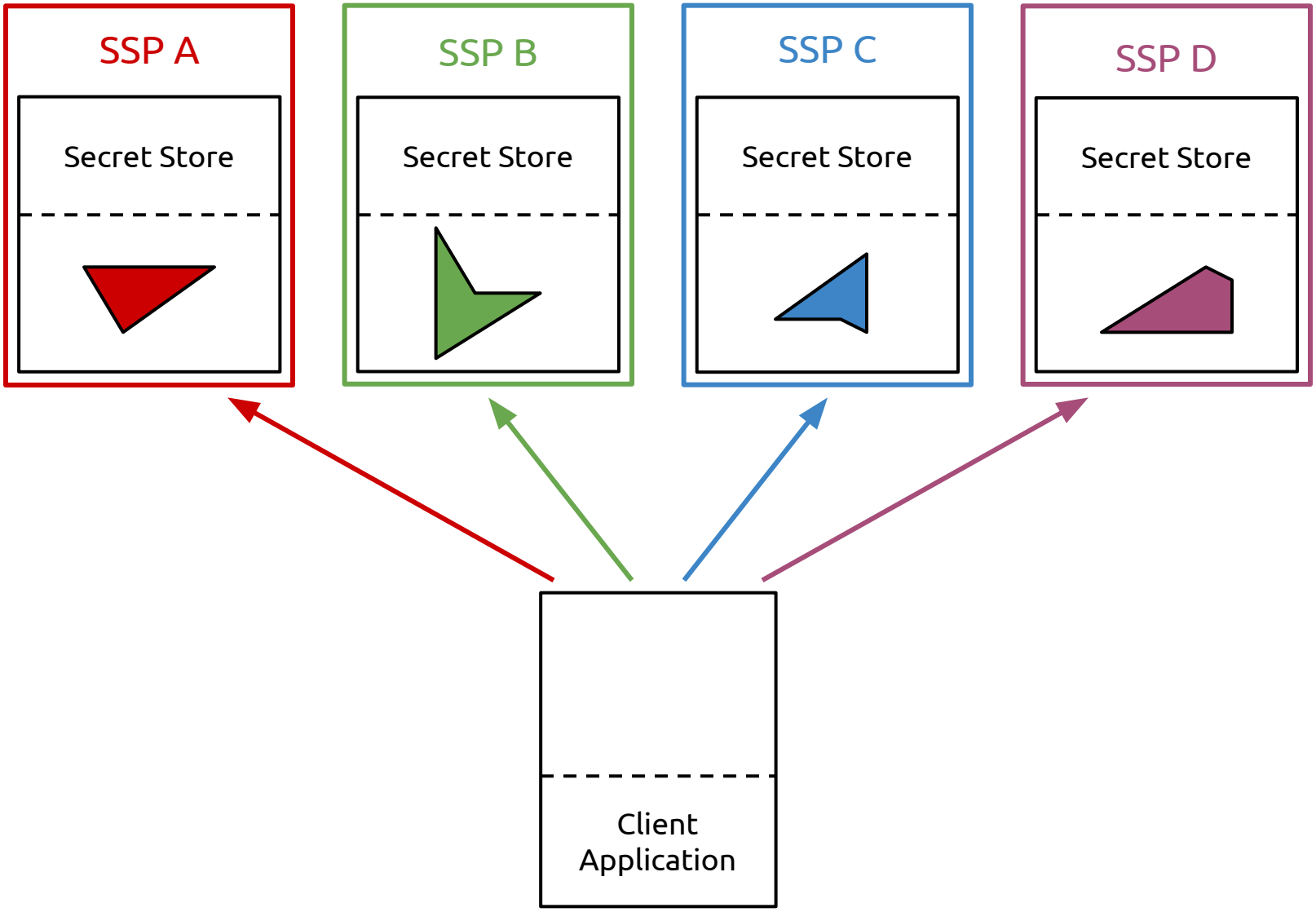

Private
Secret
Client
Application

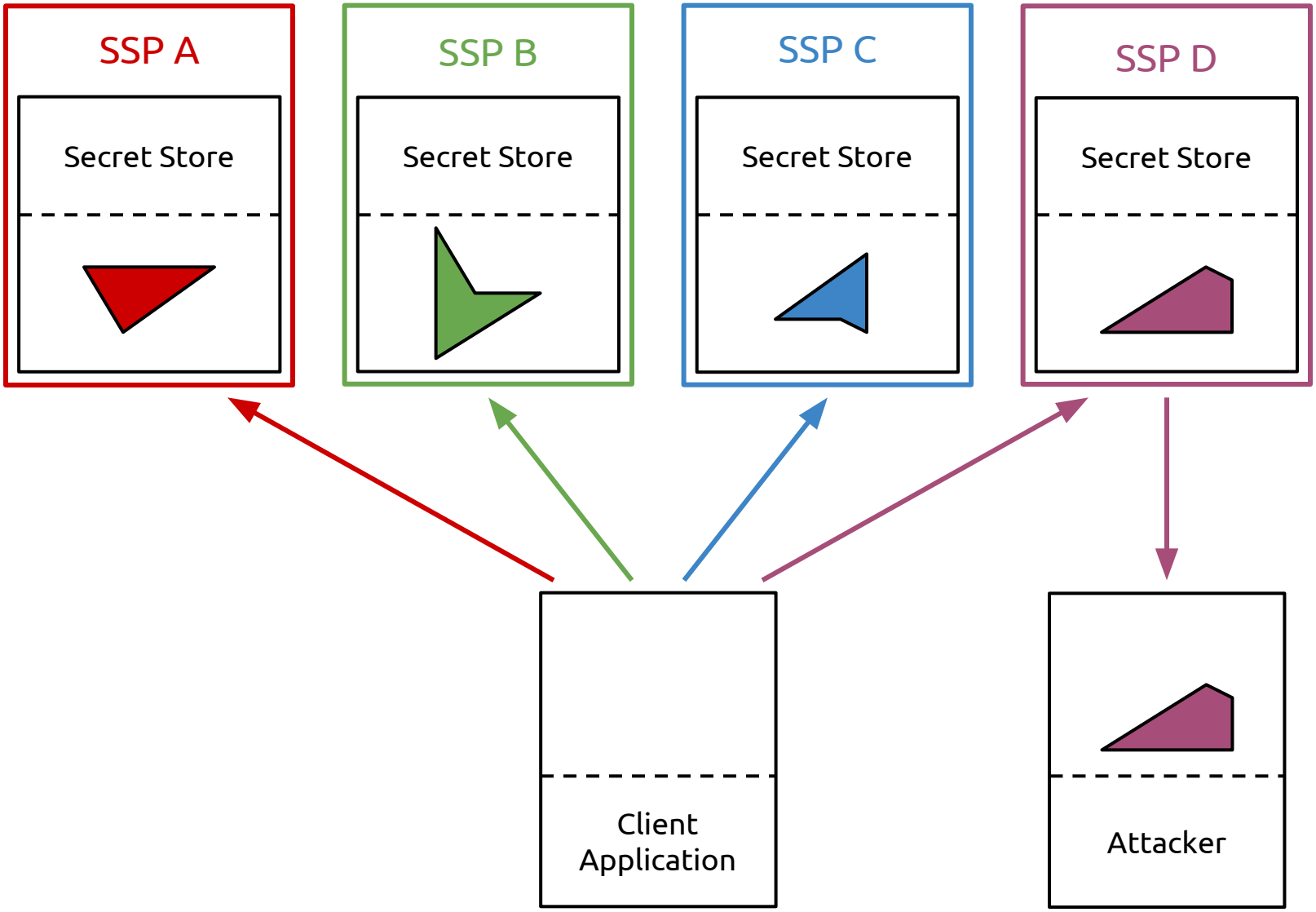

Client
Application

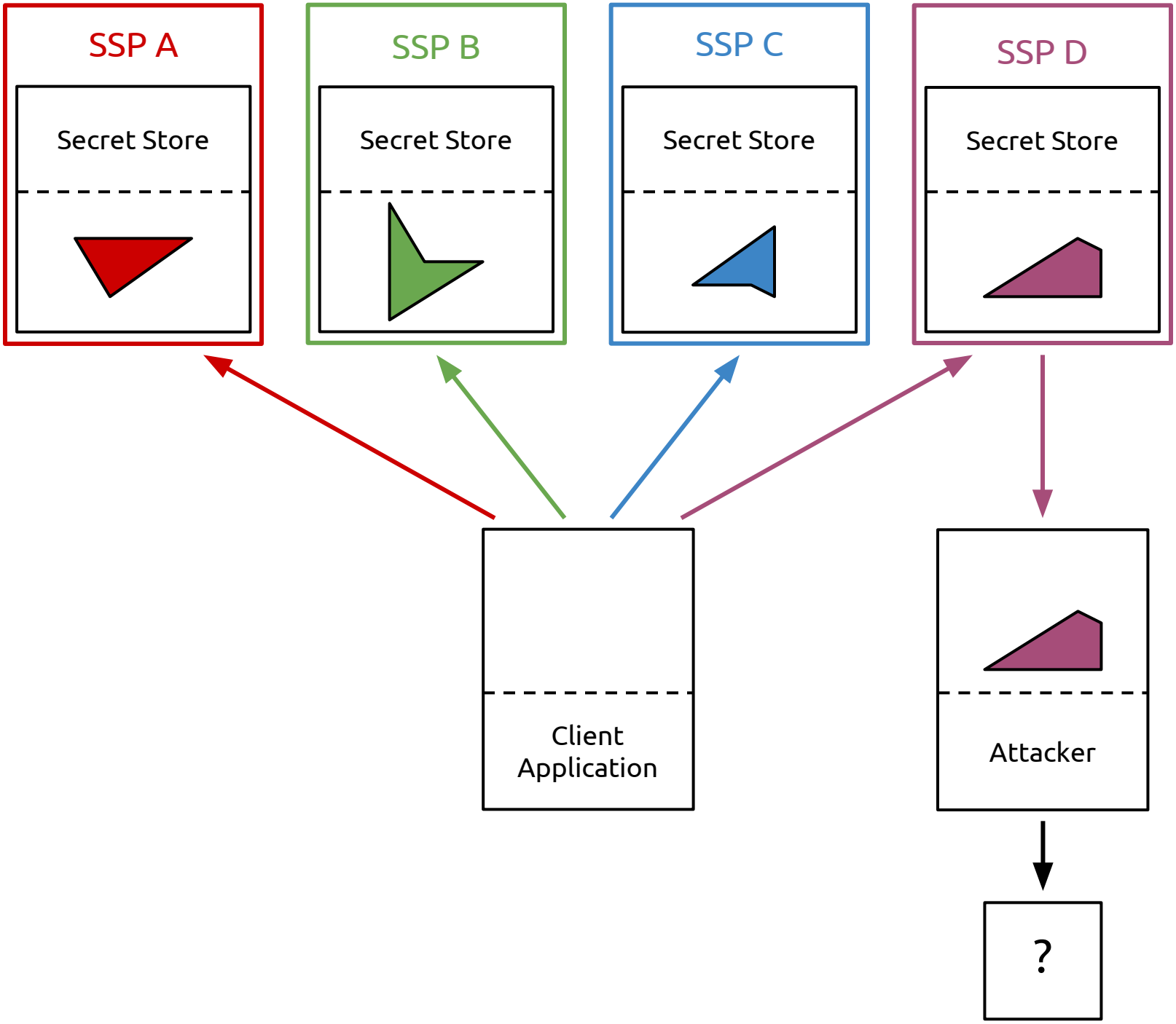

SSP A
Secret Store
Client
Application


SSP A
Secret Store
SSP B
Secret Store
Client
Application

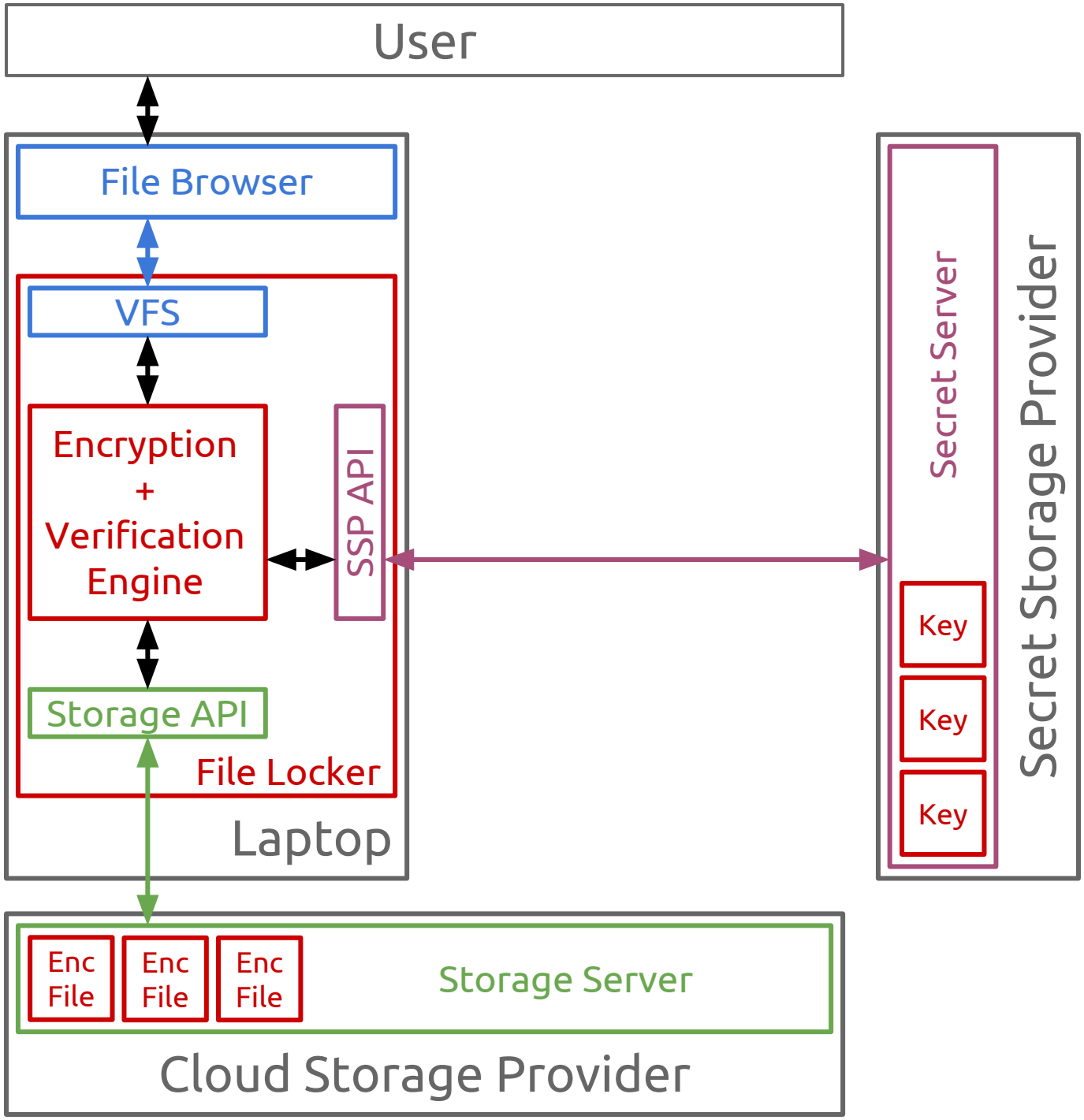

SSP A
Secret Store
SSP B
Secret Store
SSP C
Secret Store
Client
Application

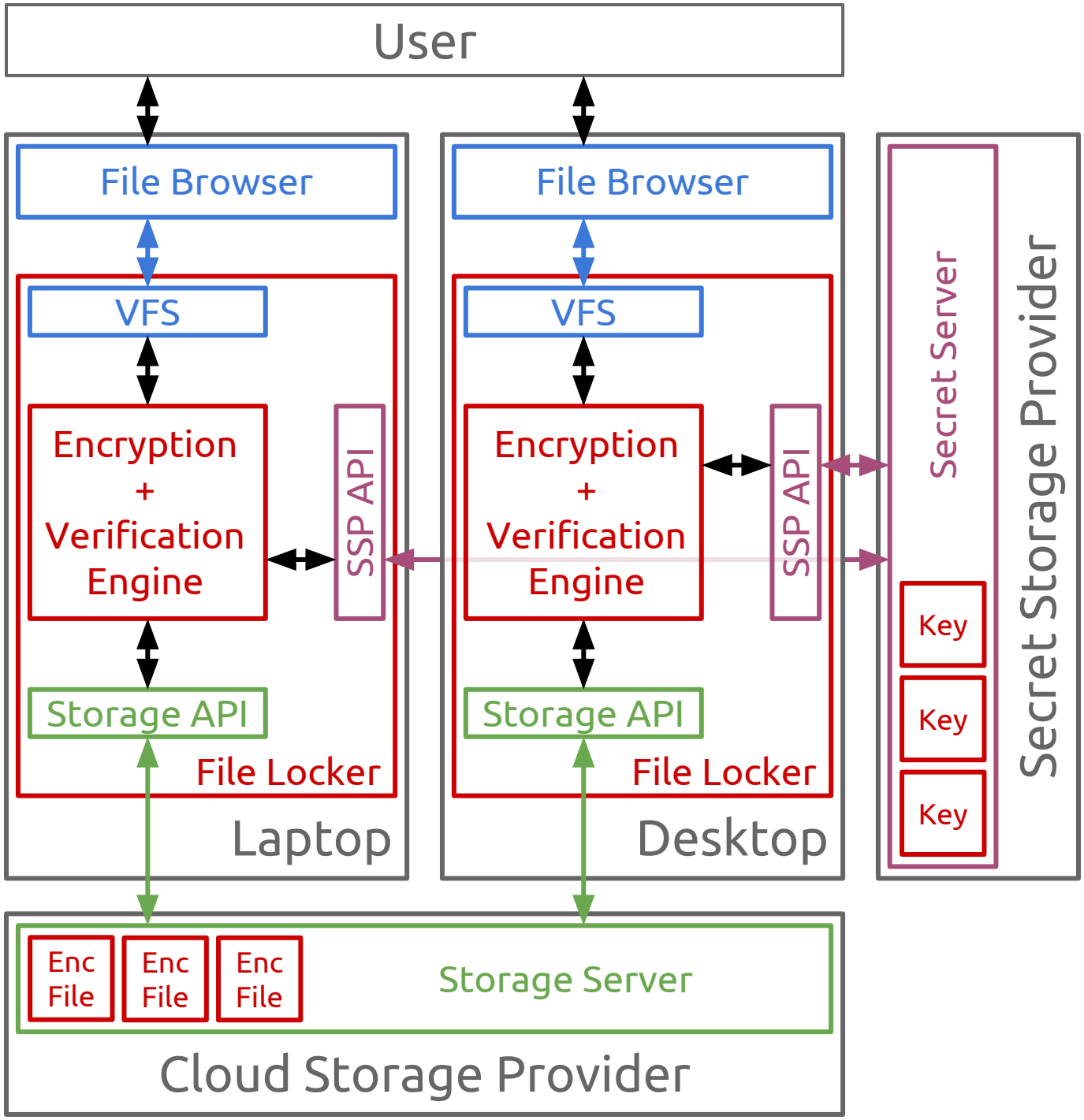

SSP A
Secret Store
SSP B
Secret Store
SSP C
Secret Store
SSP D
Secret Store
Client
Application


SSP A
Secret Store
SSP B
Secret Store
SSP C
Secret Store
SSP D
Secret Store
Client Application
Attacker

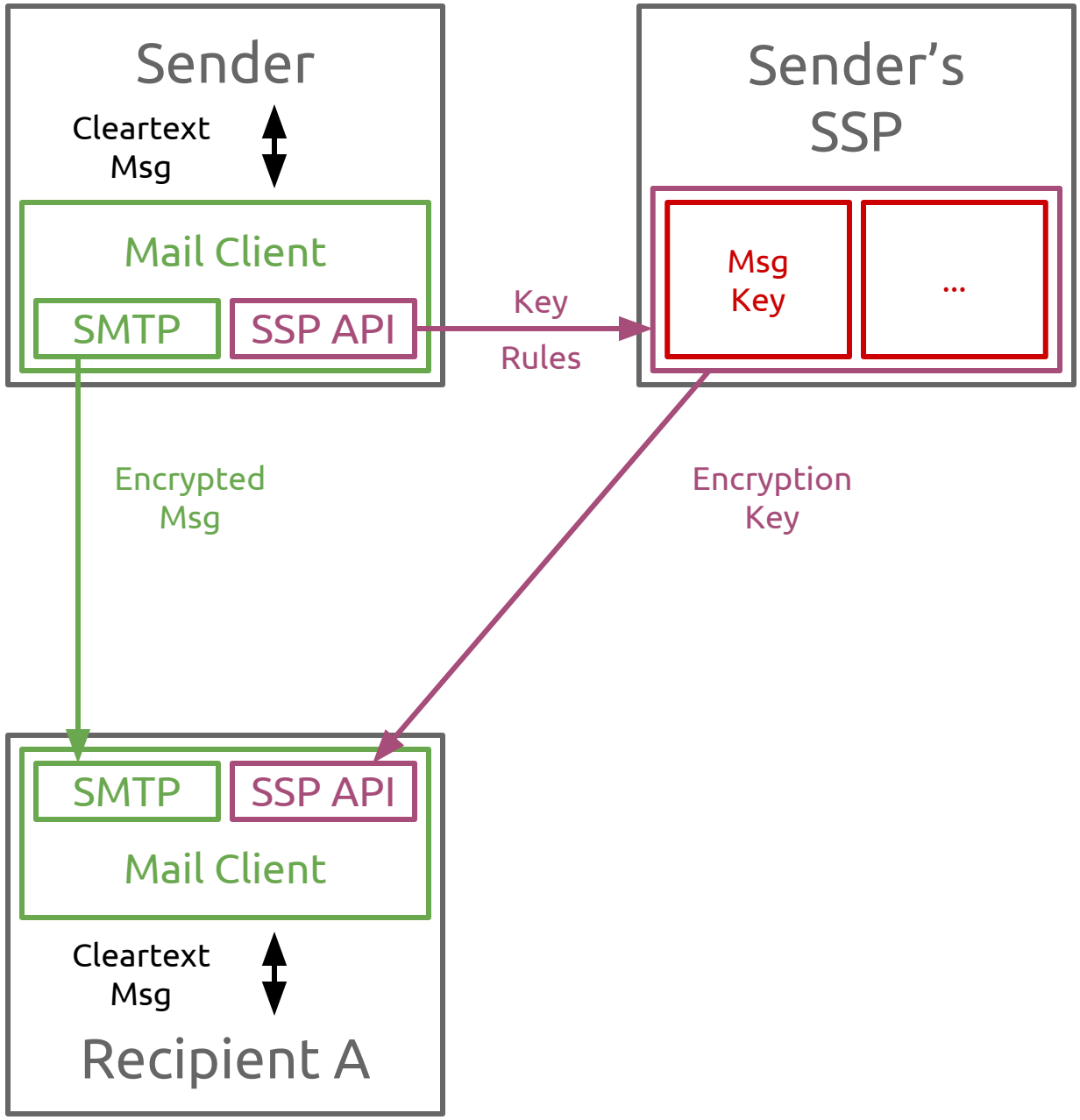

SSP A
Secret Store
SSP B
Secret Store
SSP C
Secret Store
SSP D
Secret Store
Client Application
Attacker
?

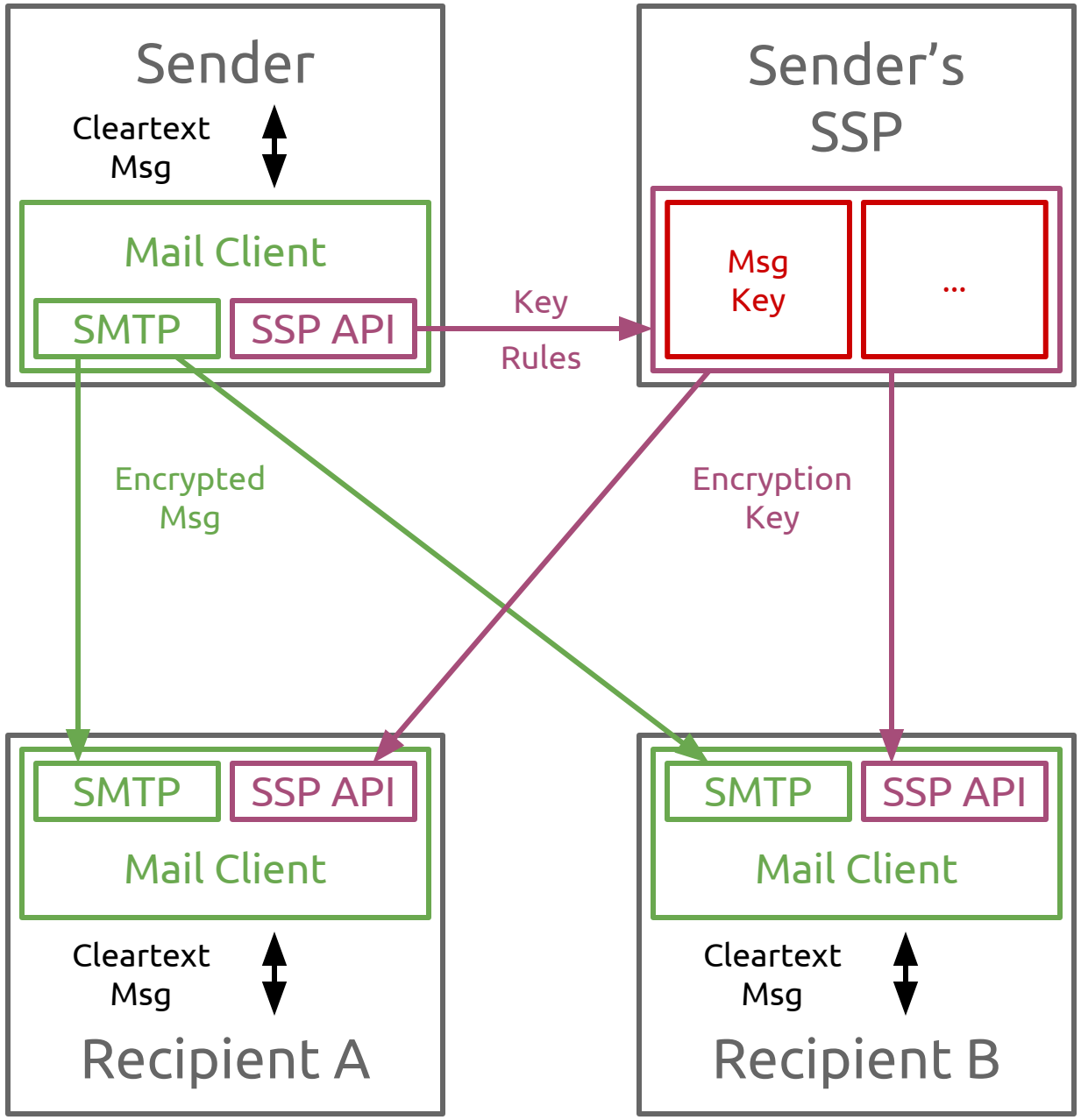
# Storage Applications


User
File Browser
VFS
Encryption
+
Verification
Engine
SSP API
Storage API
File Locker
Laptop
Secret Server
Key
Key
Key
Secret Storage Provider
Enc
File
Enc
File
Enc
File
Storage Server
Cloud Storage Provider

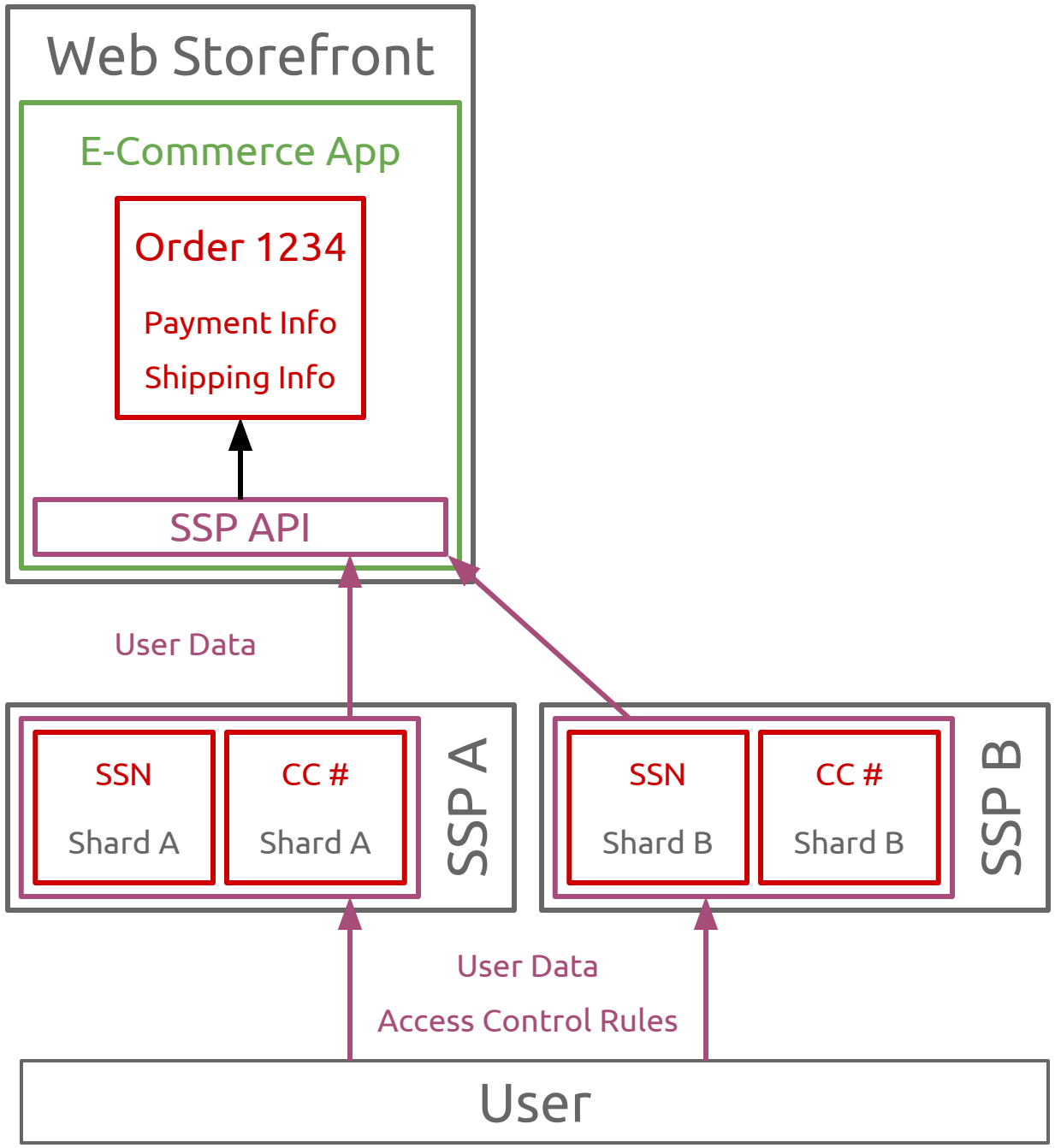

User
File Browser
VFS
Encryption
+
Verification
Engine
SSP API
Storage API
File Locker
Laptop
File Browser
VFS
Encryption
+
Verification
Engine
SSP API
Storage API
File Locker
Desktop
Secret Server
Key
Key
Key
Secret Storage Provider
Enc
File
Enc
File
Enc
File
Storage Server
Cloud Storage Provider

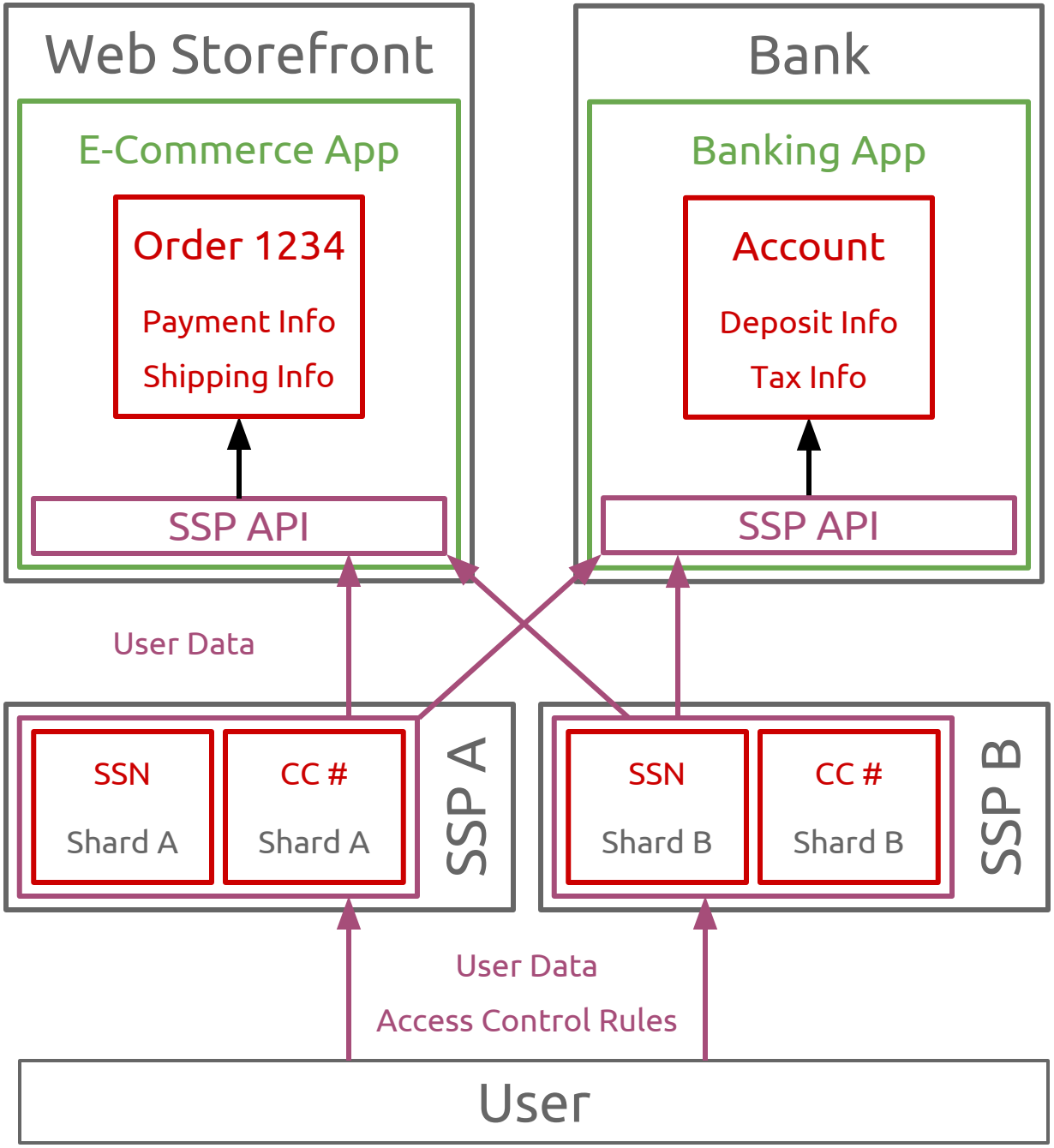
# Communication Applications


Sender
Cleartext
Msg
Mail Client
SMTP
SSP API
Sender's
SSP
Msg
Key
...
Key
Rules
Encrypted
Msg
Encryption
Key
SMTP
SSP API
Mail Client
Cleartext
Msg
Recipient A

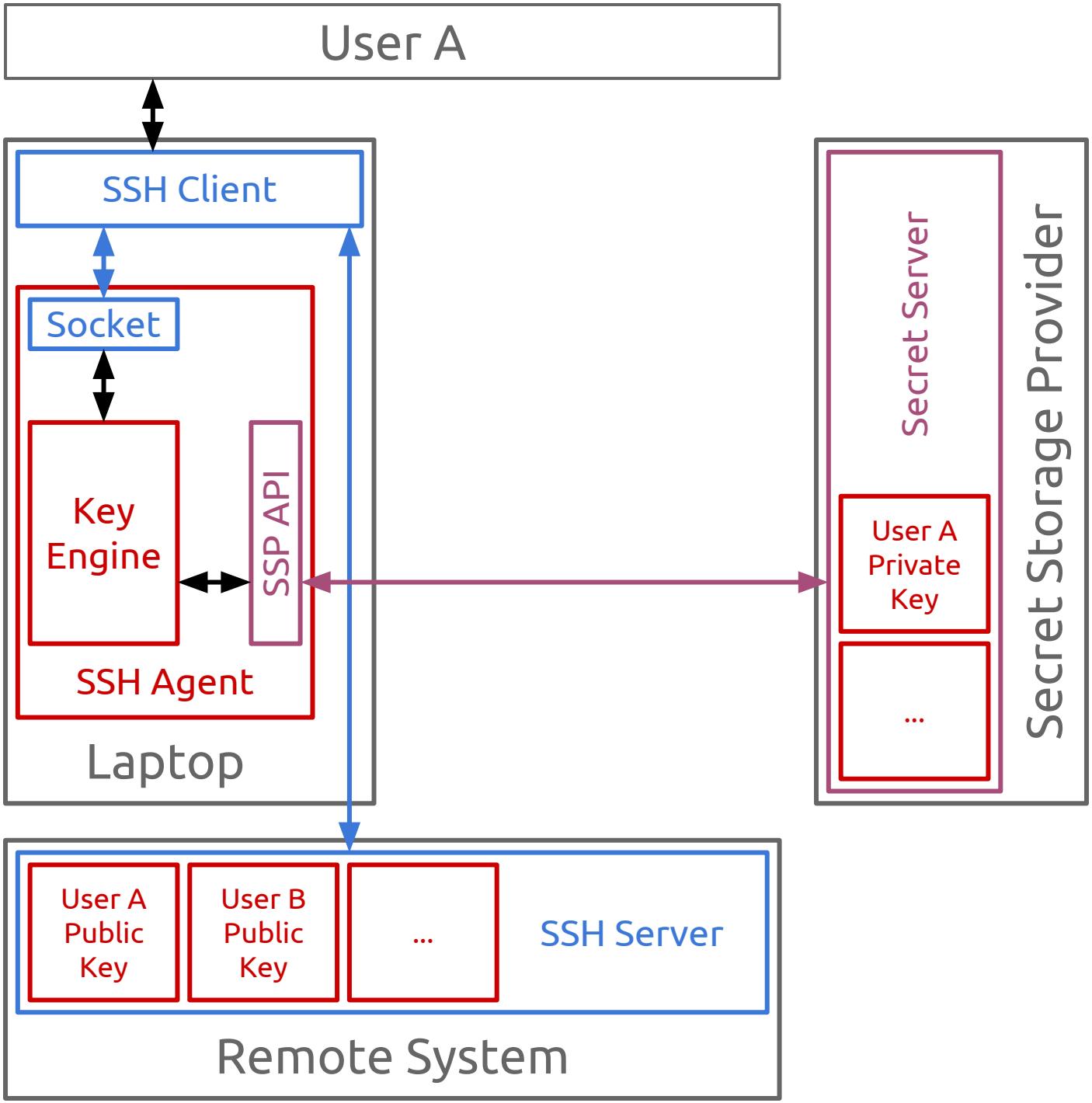

Sender
Cleartext Msg
Mail Client
SMTP
SSP API
Sender's SSP
Msg Key
...
Key
Rules
Encrypted Msg
Encryption Key
SMTP
SSP API
Mail Client
Cleartext Msg
Recipient A
SMTP
SSP API
Mail Client
Cleartext Msg
Recipient B

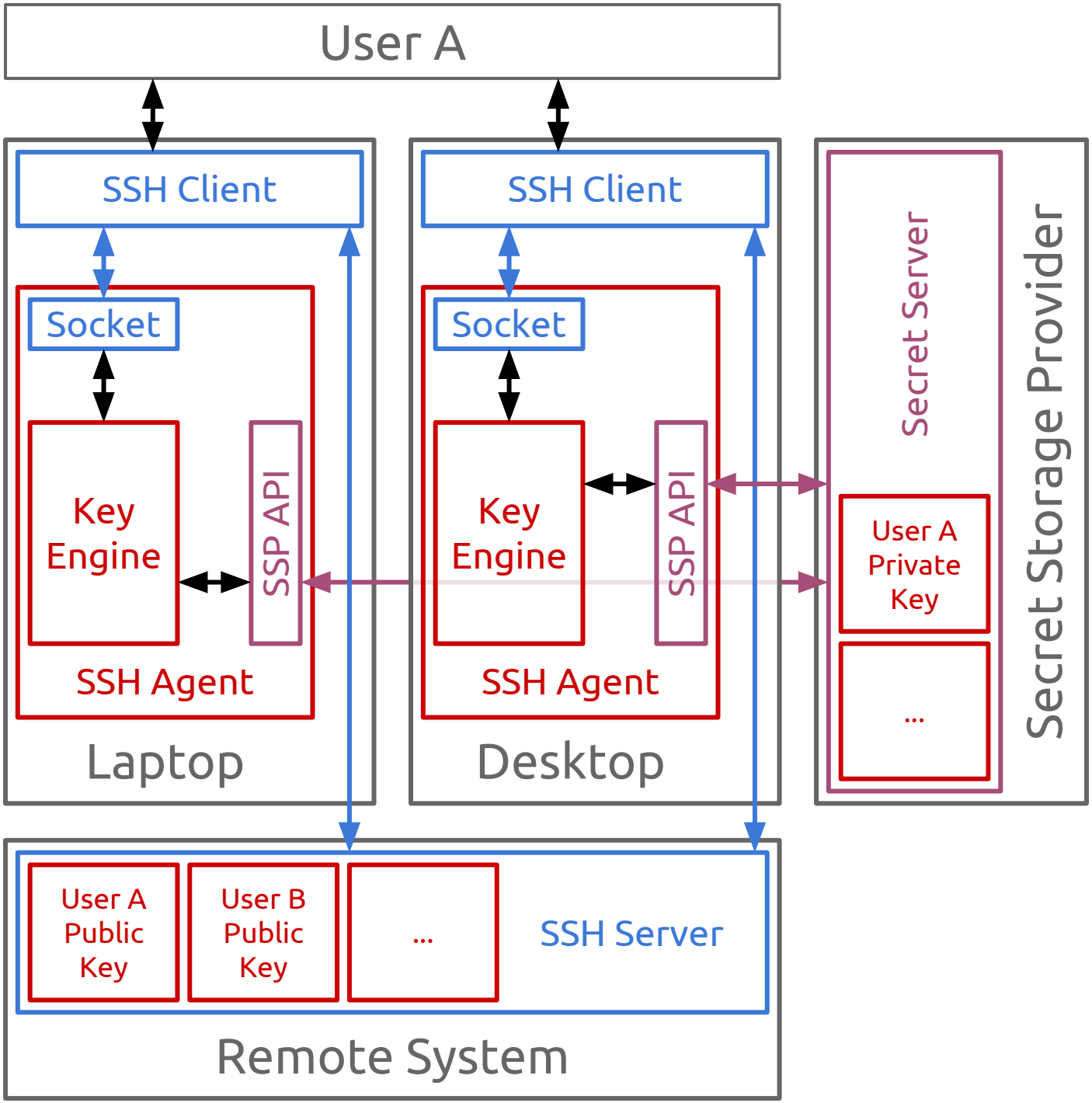
# Personal Data Repository


Web Storefront
E-Commerce App
Order 1234
Payment Info
Shipping Info
SSP API
User Data
SSN
Shard A
CC #
Shard A
SSP A
SSN
Shard B
CC #
Shard B
SSP B
User Data
Access Control Rules
User

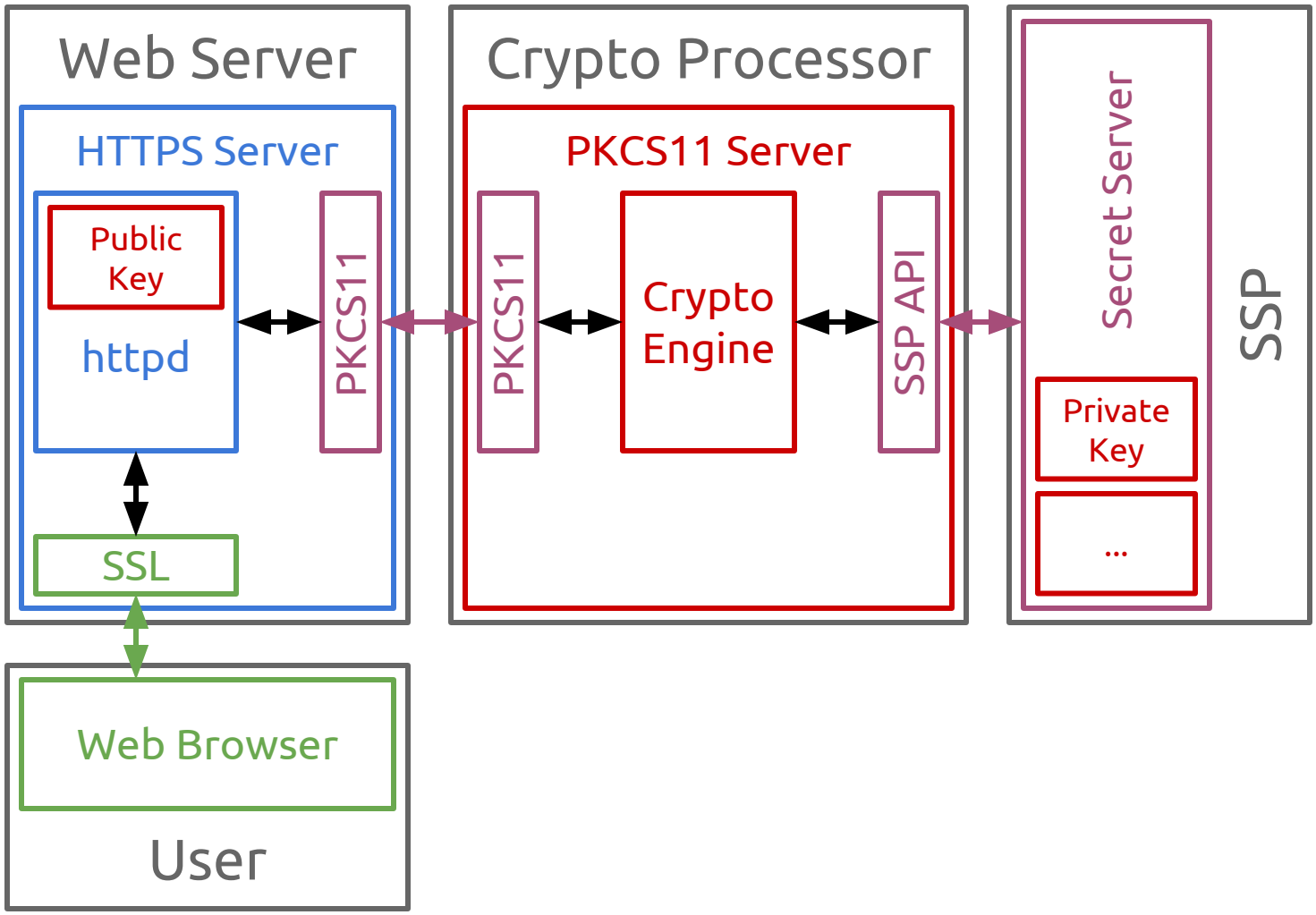

Web Storefront
E-Commerce App
Order 1234
Payment Info
Shipping Info
SSP API
Bank
Banking App
Account
Deposit Info
Tax Info
SSP API
User Data
SSN
Shard A
CC #
Shard A
SSP A
SSN
Shard B
CC #
Shard B
SSP B
User Data
Access Control Rules
User

# Authentication Applications

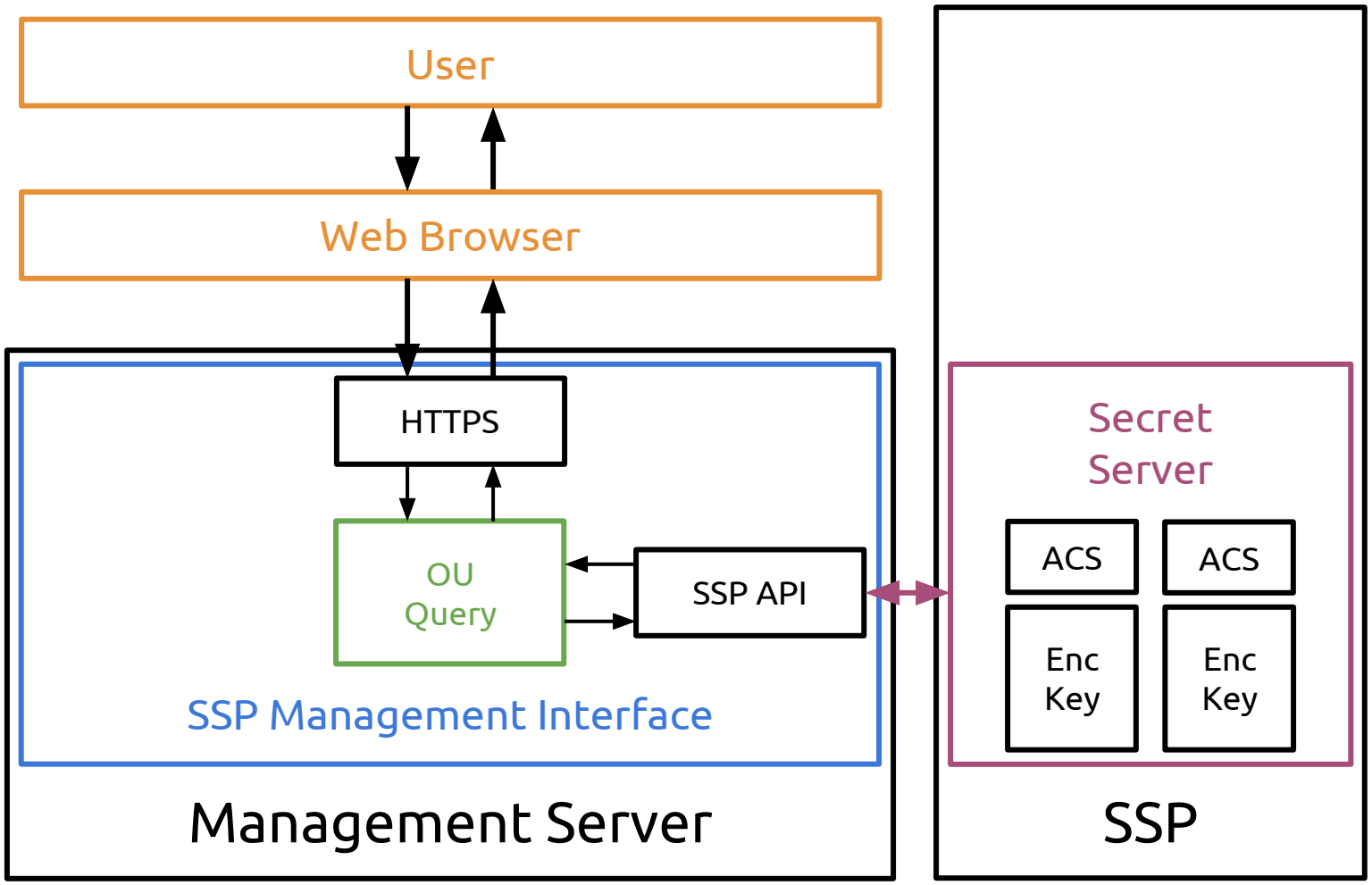

User A
SSH Client
Socket
Key Engine
SSP API
SSH Agent
Laptop
Secret Server
User A Private Key
...
Secret Storage Provider
User A Public Key
User B Public Key
...
SSH Server
Remote System


User A
SSH Client
SSH Client
Socket
Socket
Key Engine
SSP API
Key Engine
SSP API
SSH Agent
SSH Agent
Laptop
Desktop
Secret Server
User A Private Key
...
Secret Storage Provider
User A Public Key
User B Public Key
...
SSH Server
Remote System

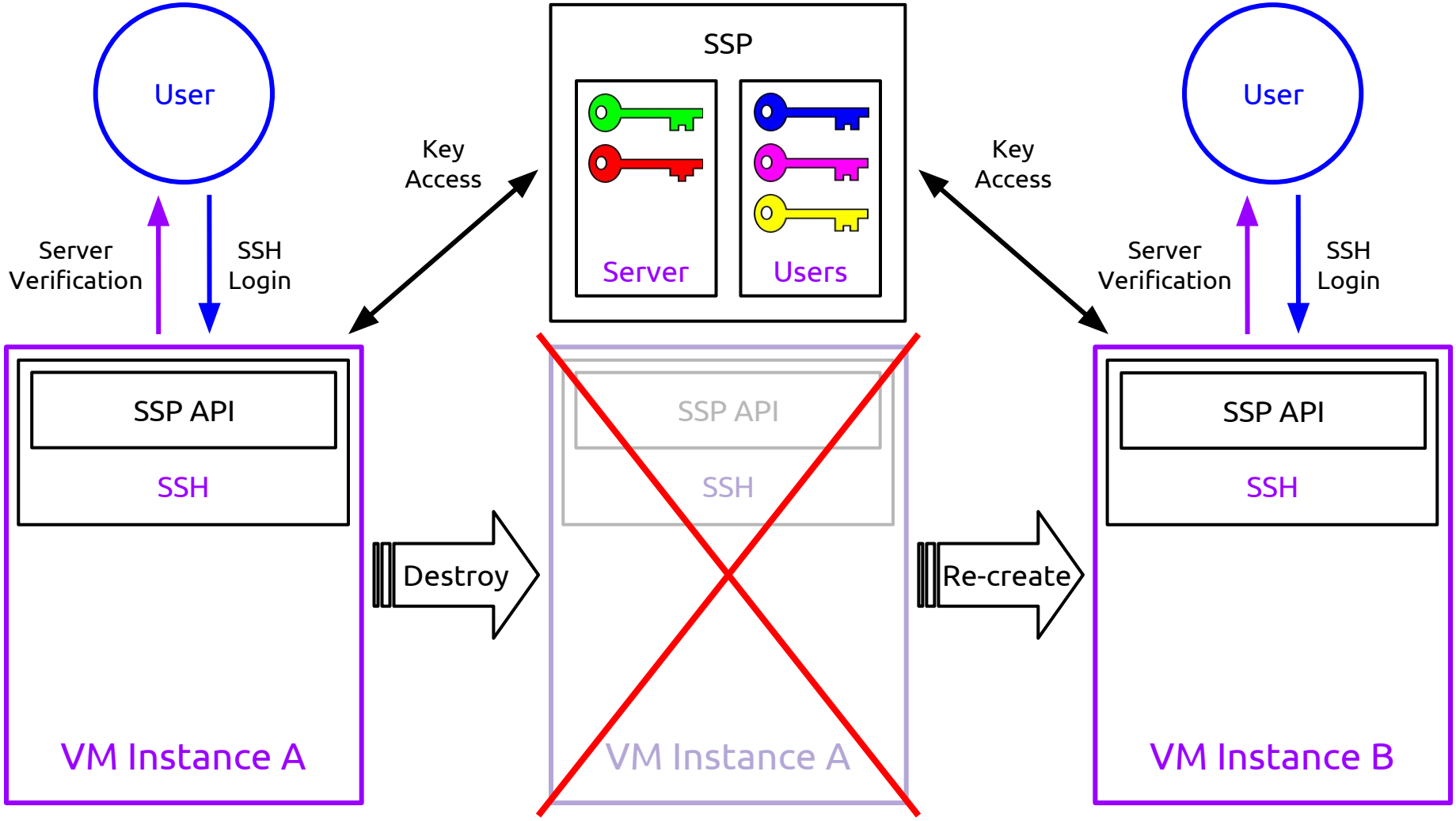
# Crypto Processing Applications


Web Server
HTTPS Server
Public Key
httpd
SSL
PKCS11
Web Browser
User
Crypto Processor
PKCS11 Server
PKCS11
Crypto Engine
SSP API
SSP
Secret Server
Private Key
...

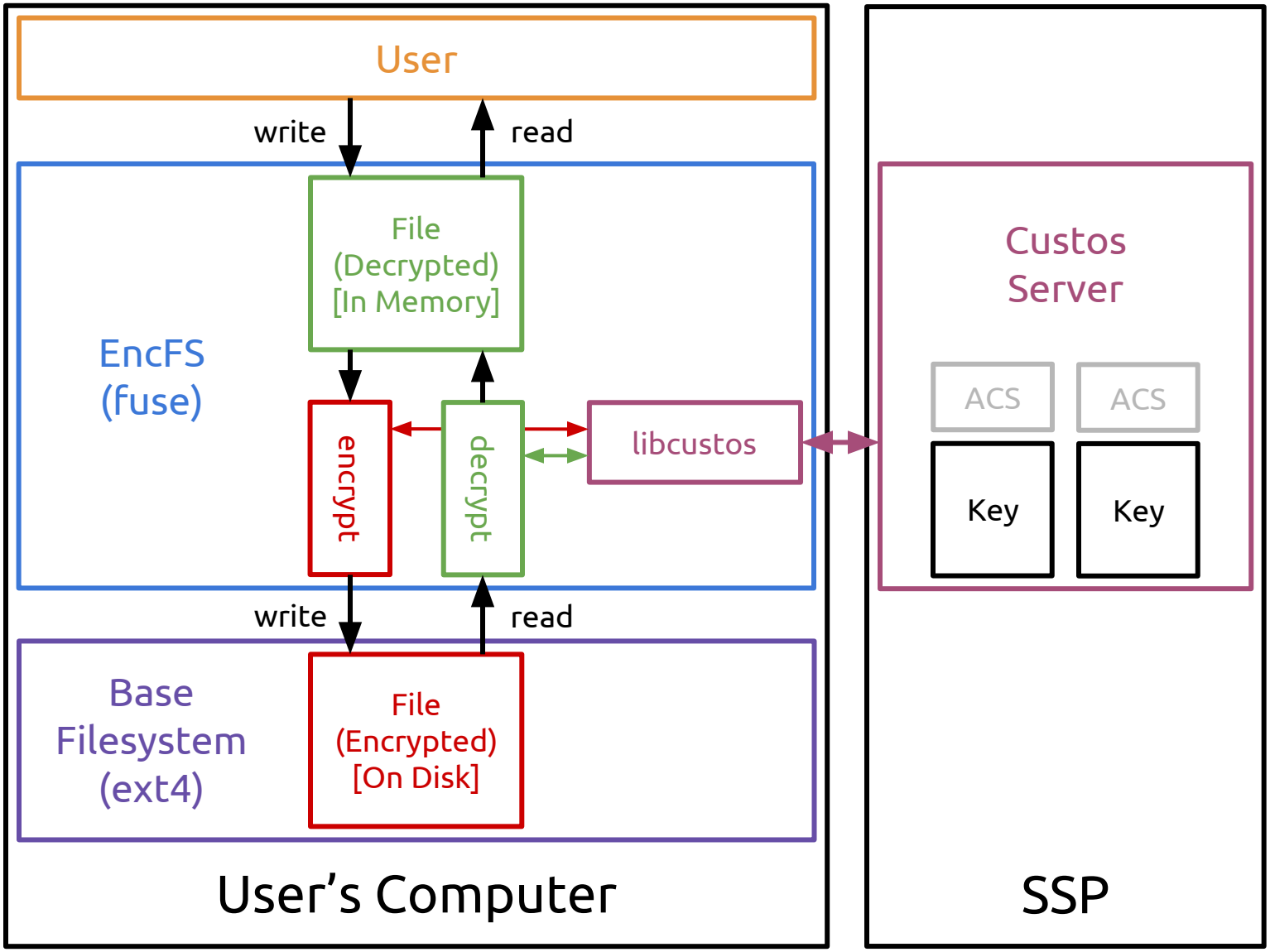
# Management Server

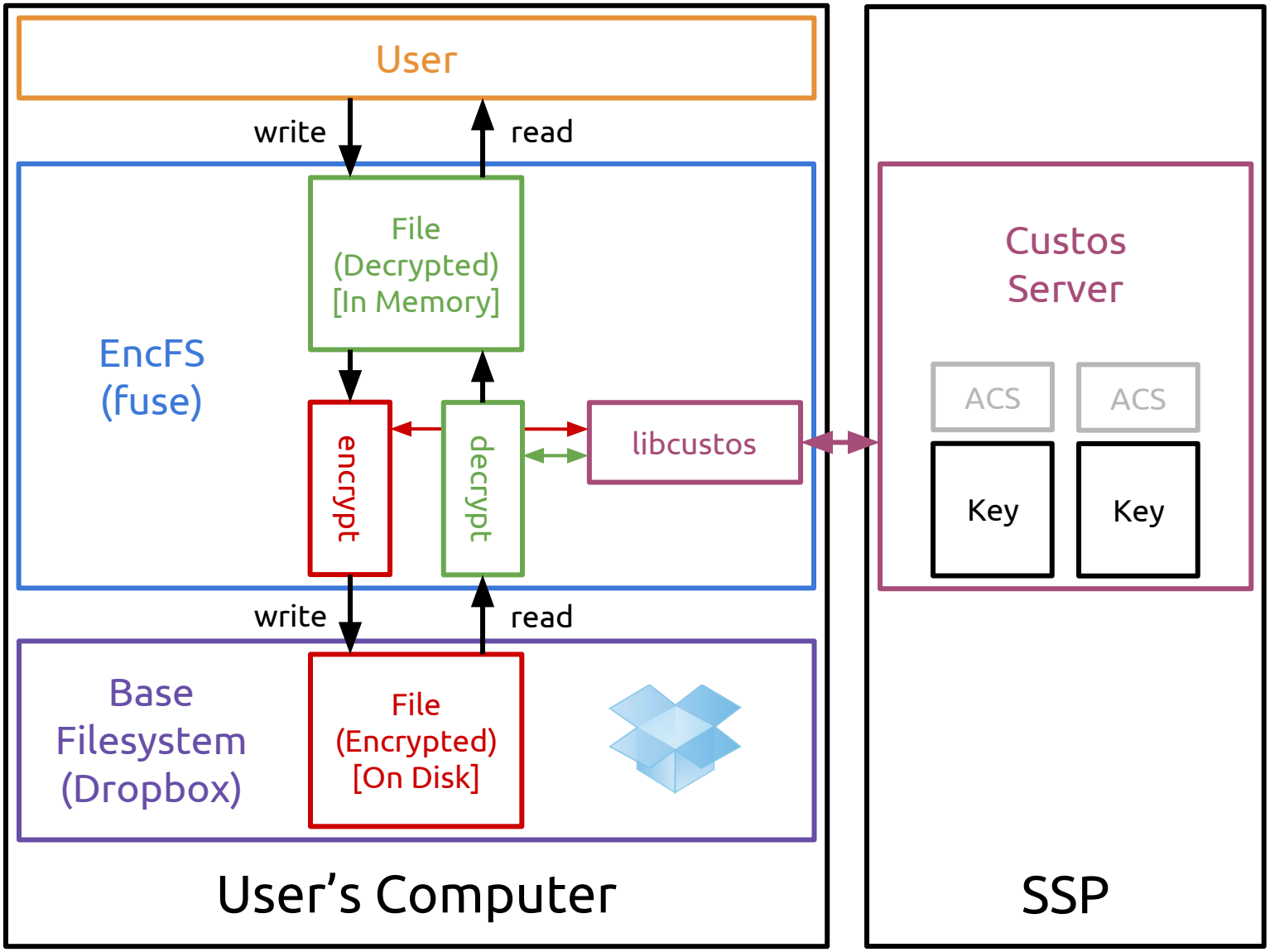

User
Web Browser
HTTPS
OU
Query
SSP API
SSP Management Interface
Management Server
Secret
Server
ACS
ACS
Enc
Key
Enc
Key
SSP

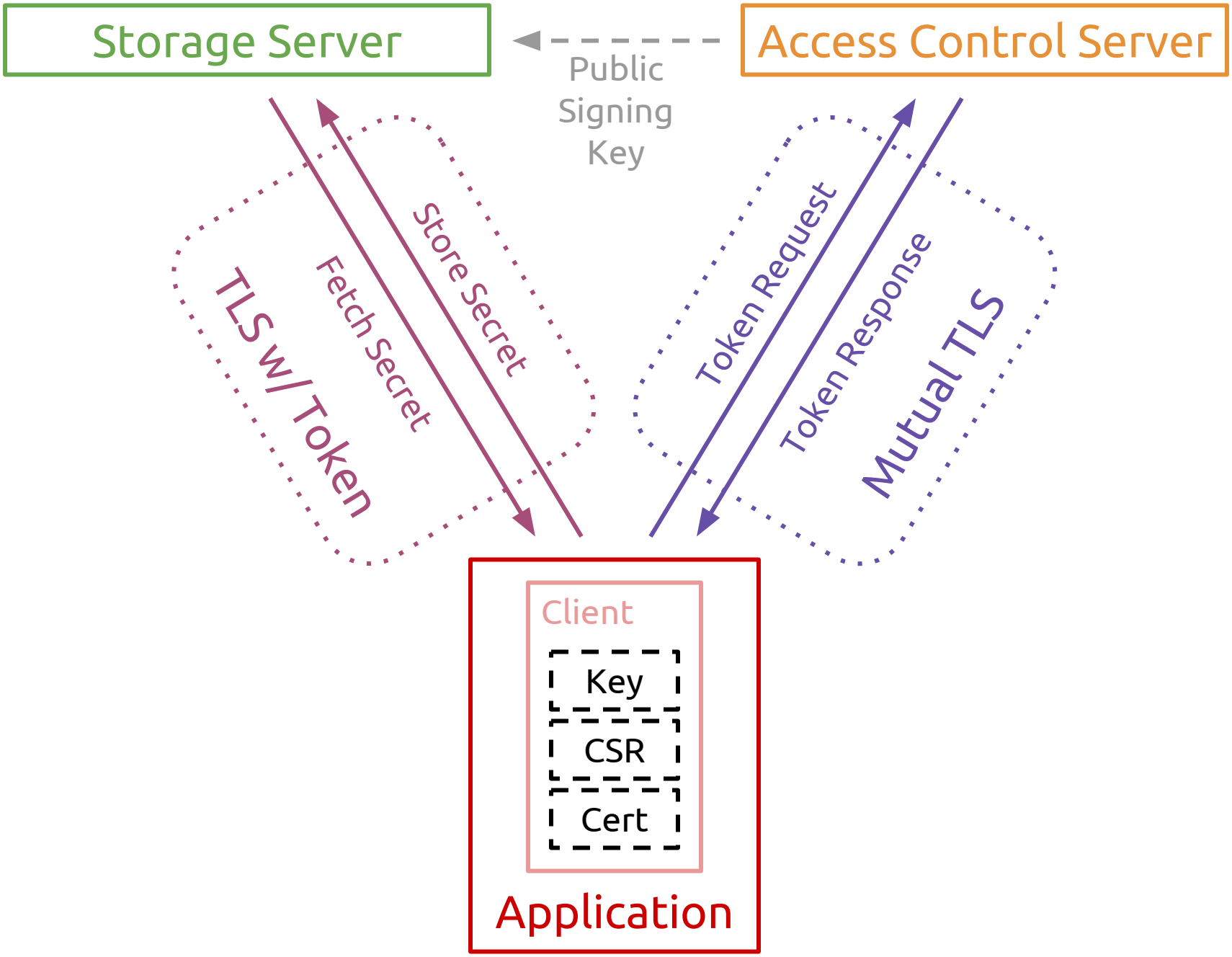
# SSH Server Key Management

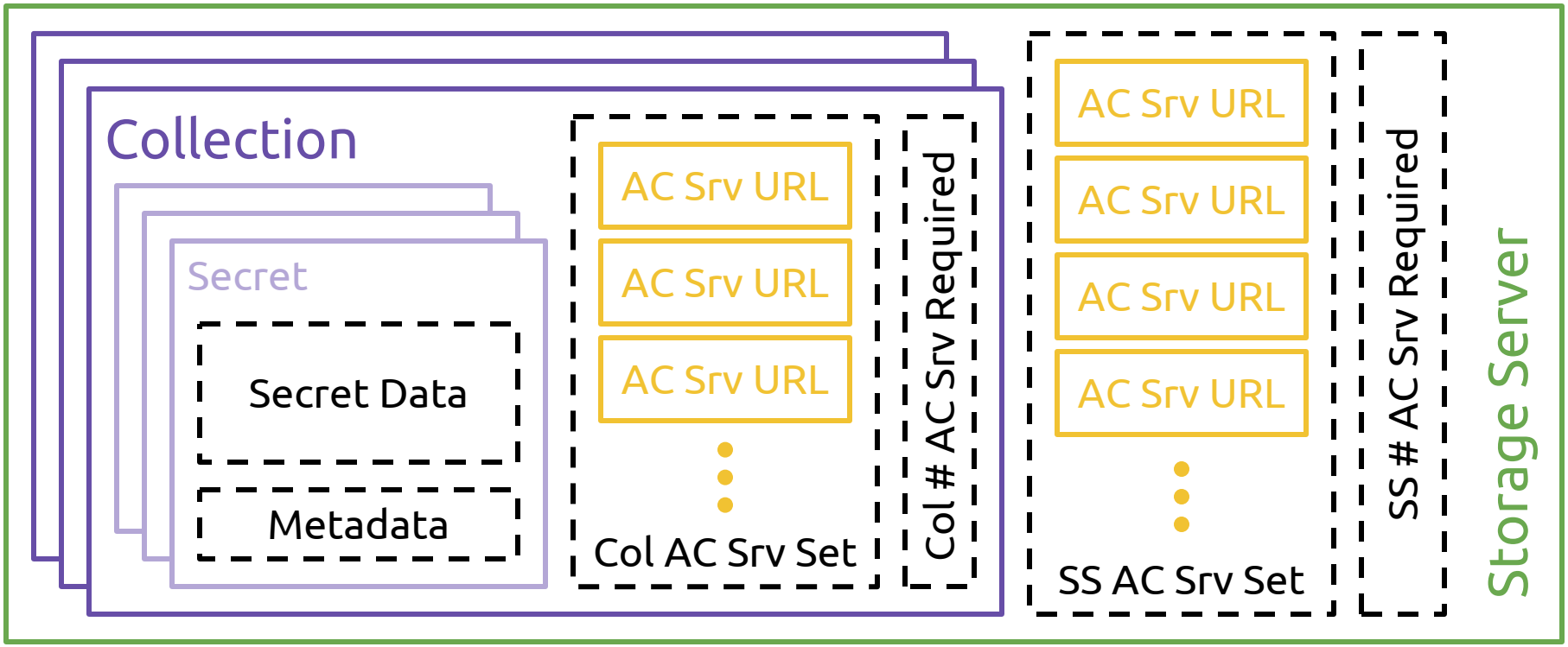

User
Server Verification
SSH Login
Key Access
SSP
Server
Users
Key Access
User
Server Verification
SSH Login
SSP API
SSH
VM Instance A
Destroy
SSP API
SSH
VM Instance A
Re-create
SSP API
SSH
VM Instance B

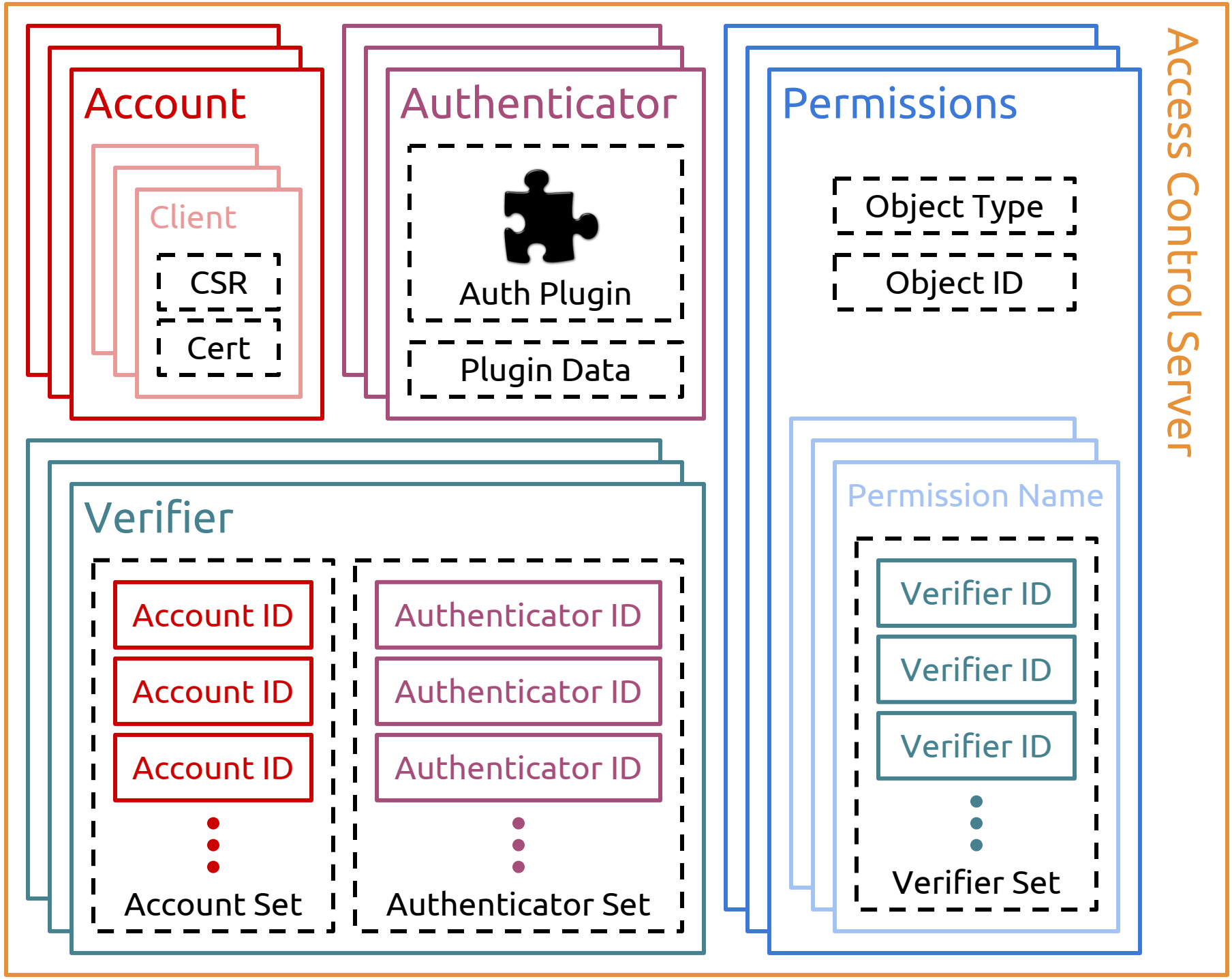
# EncFS: Custos-Backed Encrypted File System

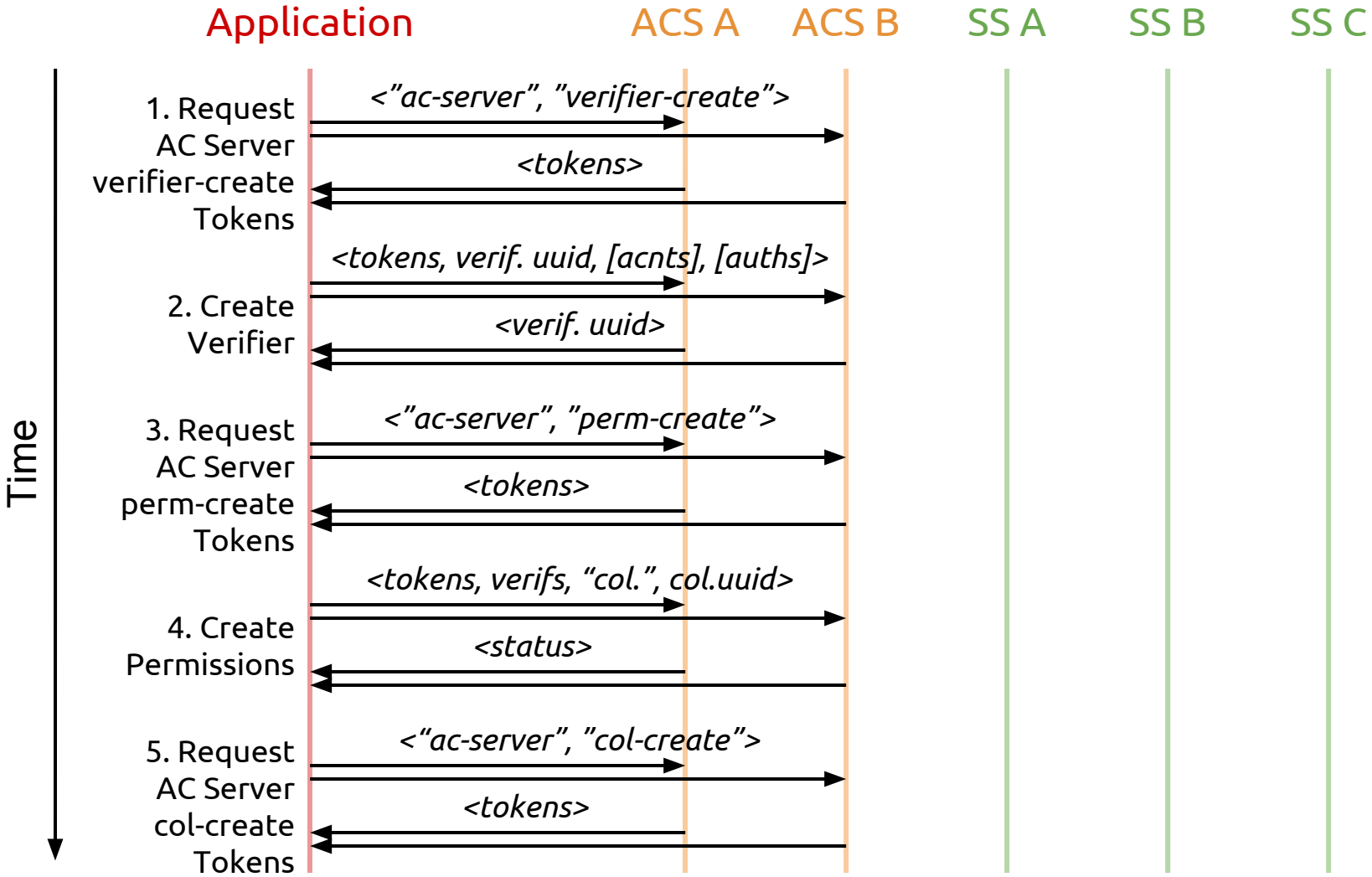

User
write
read
File
(Decrypted)
[In Memory]
EncFS
(fuse)
encrypt
decrypt
libcustos
write
read
Base
Filesystem
(ext4)
File
(Encrypted)
[On Disk]
User's Computer
Custos
Server
ACS
ACS
Key
Key
SSP

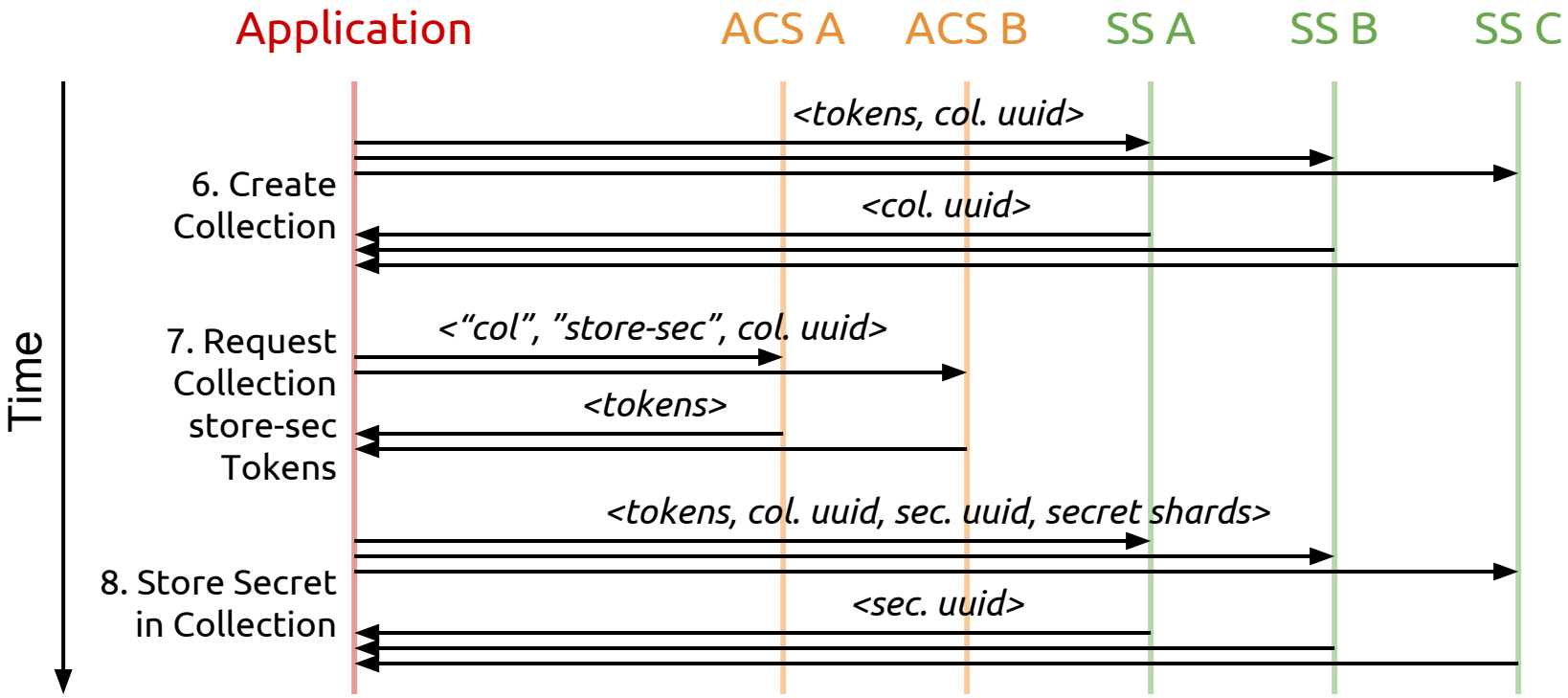

User
write
read
File
(Decrypted)
[In Memory]
EncFS
(fuse)
encrypt
decrypt
libcustos
write
read
Base
Filesystem
(Dropbox)
File
(Encrypted)
[On Disk]
User's Computer
Custos
Server
ACS
ACS
Key
Key
SSP

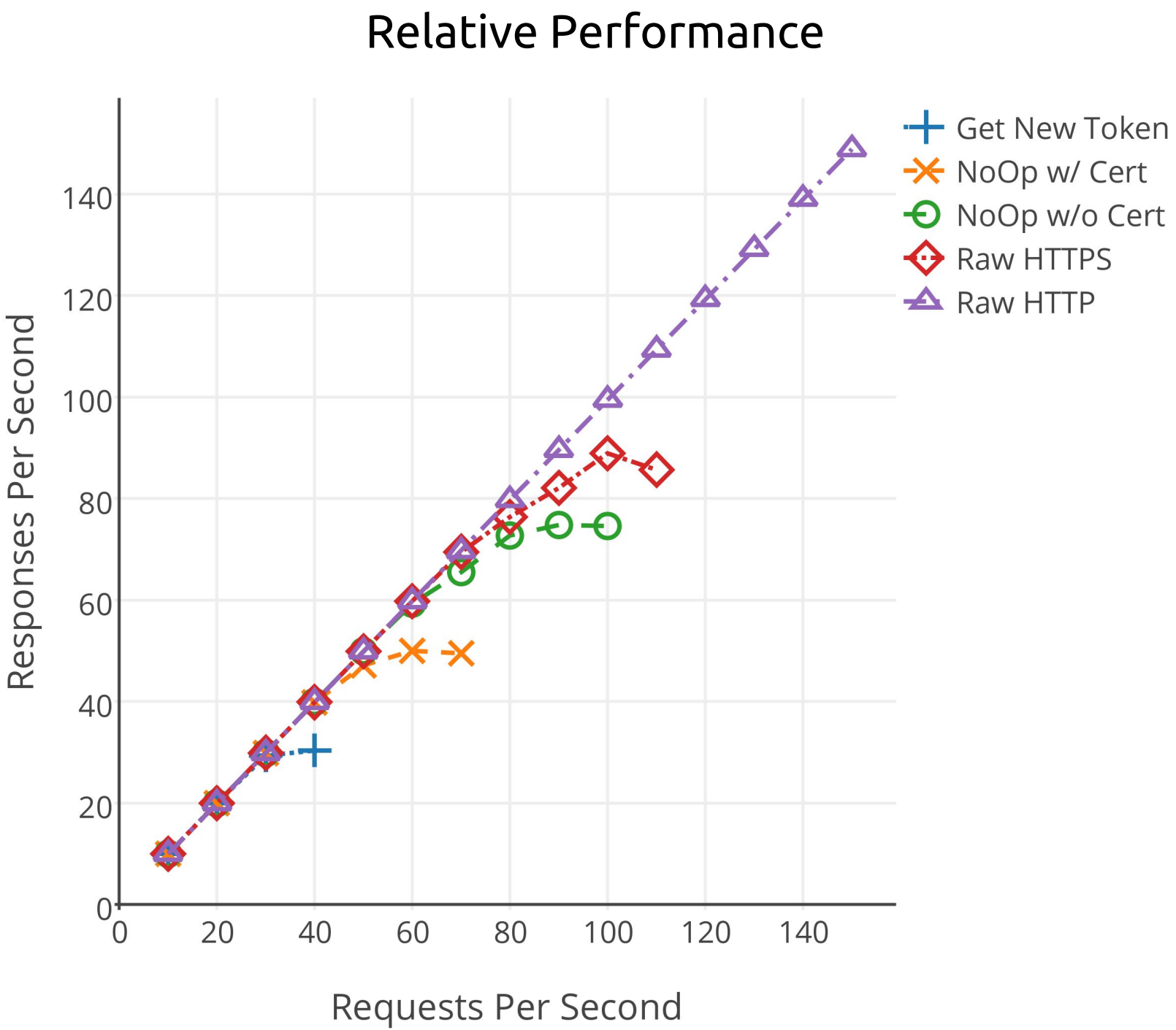

Storage Server
Access Control Server
Public Signing Key
Store Secret
Fetch Secret
TLS w/ Token
Token Request
Token Response
Mutual TLS
Client
Key
CSR
Cert
Application

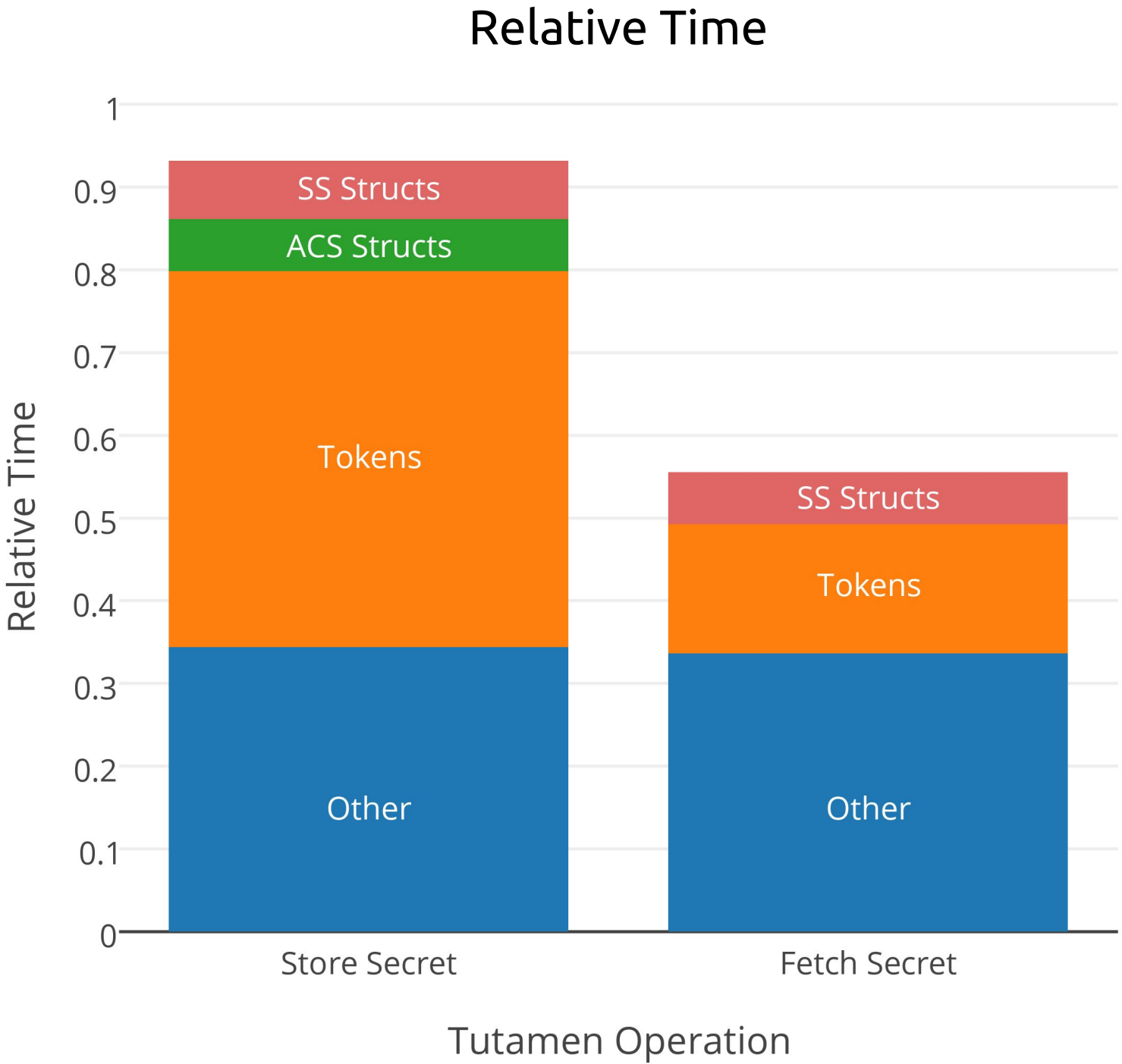

Collection
Secret
Secret Data
Metadata
AC Srv URL
AC Srv URL
AC Srv URL
Col AC Srv Set
Col # AC Srv Required
AC Srv URL
AC Srv URL
AC Srv URL
AC Srv URL
SS AC Srv Set
SS # AC Srv Required
Storage Server


Access Control Server
Account
Client
CSR
Cert
Authenticator
Auth Plugin
Plugin Data
Permissions
Object Type
Object ID
Permission Name
Verifier ID
Verifier ID
Verifier ID
Verifier Set
Verifier
Account ID
Account ID
Account ID
Account Set
Authenticator ID
Authenticator ID
Authenticator ID
Authenticator Set


Application
ACS A
ACS B
SS A
SS B
SS C
Time
1. Request AC Server verifier-create Tokens
<"ac-server", "verifier-create">
<tokens>
2. Create Verifier
<tokens, verif. uuid, [acnts], [auths]>
<verif. uuid>
3. Request AC Server perm-create Tokens
<"ac-server", "perm-create">
<tokens>
4. Create Permissions
<tokens, verifs, "col.", col.uuid>
<status>
5. Request AC Server col-create Tokens
<"ac-server", "col-create">
<tokens>


Application
ACS A
ACS B
SS A
SS B
SS C
Time
6. Create Collection
<tokens, col. uuid>
<col. uuid>
7. Request Collection store-sec Tokens
<"col", "store-sec", col. uuid>
<tokens>
8. Store Secret in Collection
<tokens, col. uuid, sec. uuid, secret shards>
<sec. uuid>


Relative Performance
Responses Per Second
Requests Per Second
0
20
40
60
80
100
120
140
Get New Token
NoOp w/ Cert
NoOp w/o Cert
Raw HTTPS
Raw HTTP


Relative Time
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Relative Time
SS Structs
ACS Structs
Tokens
Other
SS Structs
Tokens
Other
Store Secret
Fetch Secret
Tutamen Operation

# Future Work

# Auditing -> Automation

Auditing -> Automation

Performance

Auditing -> Automation

Performance

Additional Client Integrations