



**Infinite power of the web !**



# Infinite power of the web !

- Share anything, anytime



# Infinite power of the web !

- Share anything, anytime
- Real time data



# Infinite power of the web !

- Share anything, anytime
- Real time data
- Unlimited storage (and knowledge!)





# Infinite power of the web !

- Share anything, anytime
- Real time data
- Unlimited storage (and knowledge!)
- Cloud computing power



# Infinite power of the web !

- Share anything, anytime
- Real time data
- Unlimited storage (and knowledge!)
- Cloud computing power
- LLM



# Infinite power of the web !

- Share anything, anytime
- Real time data
- Unlimited storage (and knowledge!)
- Cloud computing power
- LLM
- Decentralized backups
- ...

# Web API Security

Ensure your app is safe!

Mobilis in Mobile 2024

# Thomas Durand

Call me Dean



<https://thomasdurand.fr>

[@deanatoire@mastodon.social](mailto:deanatoire@mastodon.social)

[@deanatoire@threads.net](mailto:deanatoire@threads.net)

[@deanatoire@twitter.com](https://twitter.com/deanatoire)

# Thomas Durand

Call me Dean



Graduate from Centrale Nantes

<https://thomasdurand.fr>

[@deanatoire@mastodon.social](mailto:@deanatoire@mastodon.social)

[@deanatoire@threads.net](mailto:@deanatoire@threads.net)

[@deanatoire@twitter.com](mailto:@deanatoire@twitter.com)



# Thomas Durand

Call me Dean



Graduate from Centrale Nantes

Backend architect at DiliTrust

<https://thomasdurand.fr>

[@deanatoire@mastodon.social](mailto:@deanatoire@mastodon.social)

[@deanatoire@threads.net](mailto:@deanatoire@threads.net)

[@deanatoire@twitter.com](mailto:@deanatoire@twitter.com)

# Thomas Durand

Call me Dean



Graduate from Centrale Nantes

Backend architect at DiliTrust

iOS indie dev



<https://thomasdurand.fr>

[@deanatoire@mastodon.social](mailto:@deanatoire@mastodon.social)

[@deanatoire@threads.net](mailto:@deanatoire@threads.net)

[@deanatoire@twitter.com](mailto:@deanatoire@twitter.com)



# Thomas Durand

Call me Dean



Graduate from Centrale Nantes

Backend architect at DiliTrust

iOS indie dev



Speaker, tech blog writer,  enthusiast

<https://thomasdurand.fr>

[@deanatoire@mastodon.social](mailto:deanatoire@mastodon.social)

[@deanatoire@threads.net](mailto:deanatoire@threads.net)

[@deanatoire@twitter.com](mailto:deanatoire@twitter.com)



Padlok



SharePal



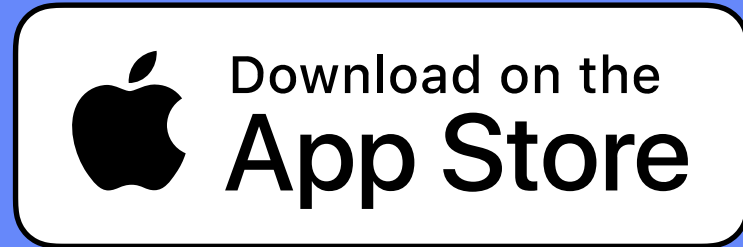
Discount today!



Padlok



SharePal



**In this session**

# In this session

Communication between your app and the web

# In this session

Communication between your app and the web

Cryptography principles

# In this session

Communication between your app and the web

Cryptography principles

Identification and Authentication

# In this session

Communication between your app and the web

Cryptography principles

Identification and Authentication

A word on customer data



# In this session

Communication between your app and the web

Cryptography principles

Identification and Authentication

A word on customer data

My cat



# Newton

## European



# Newton

## European

Sleeping



# Newton

## European

Sleeping

Eating



# Newton

## European

Sleeping

Eating

Purring



# Newton

## European

Sleeping

Eating

Purring

Ignore the dog





# Charlie

English Cocker Spaniel



# Charlie

## English Cocker Spaniel

Buddies





# Charlie

## English Cocker Spaniel



Buddies

Eating

# Charlie

## English Cocker Spaniel



Buddies

Eating

Walking

# Charlie

## English Cocker Spaniel



Buddies

Eating

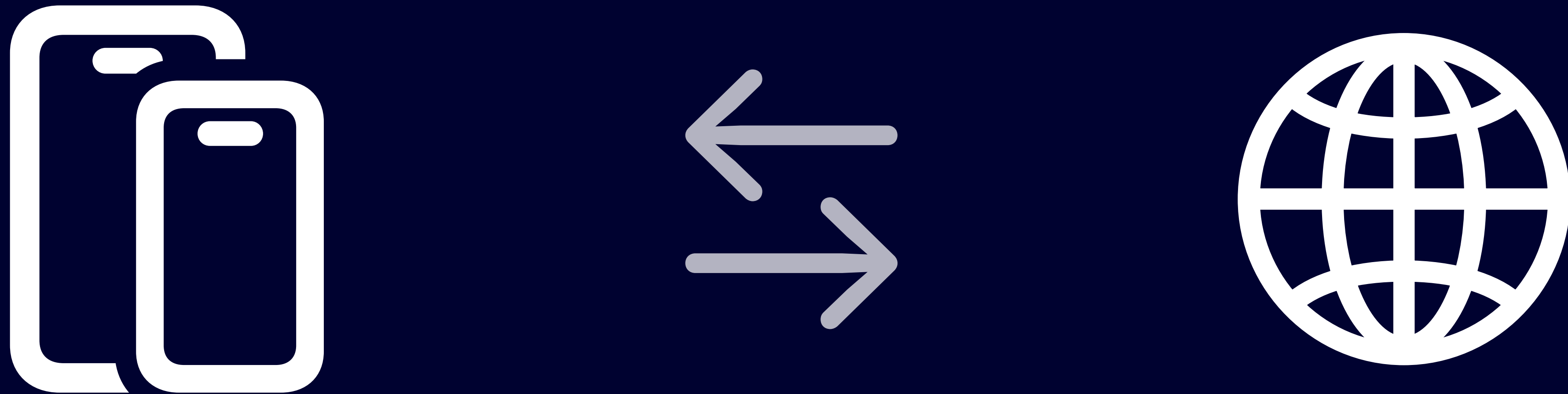
Walking

Love the cat

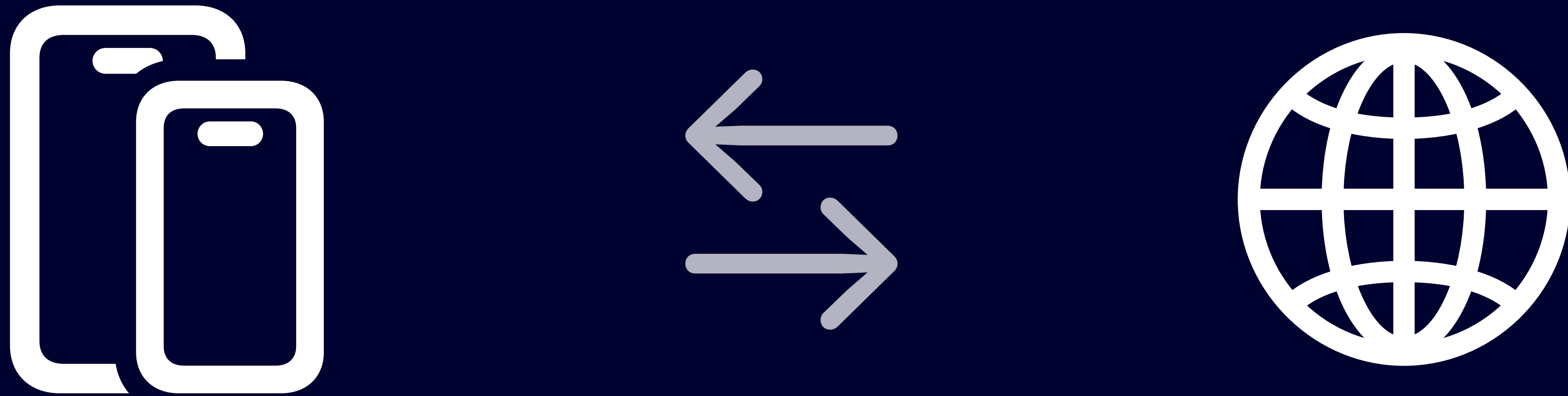


# Communication between your app and the web

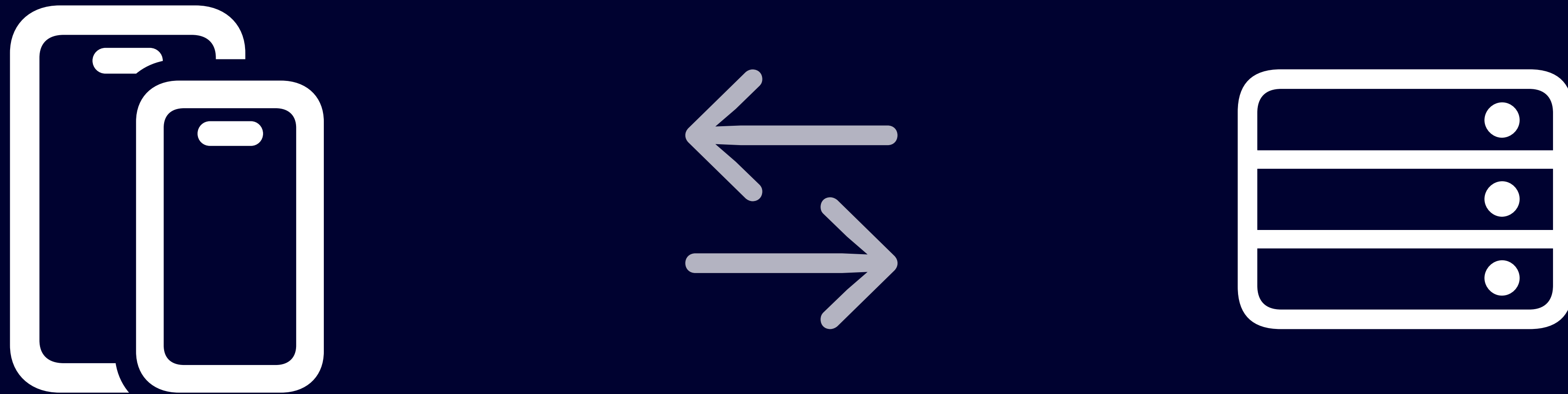
# Communication between your app and the web



# Communication between your app and server

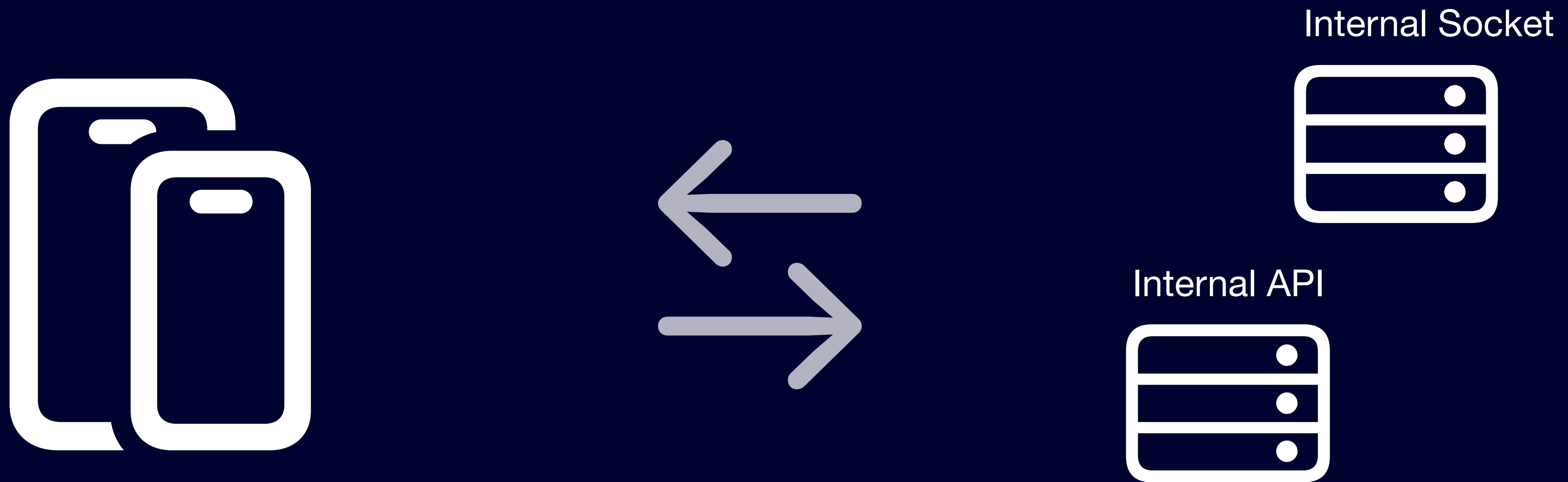


# Communication between your app and server





# Communication between your app and servers

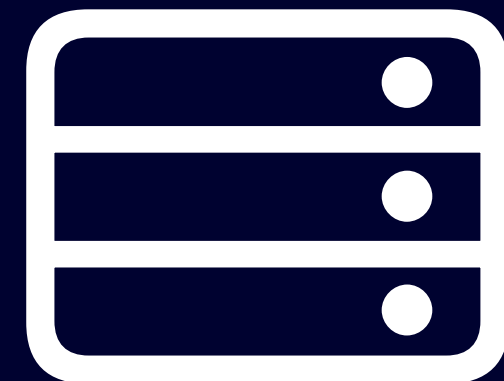


Request / Response

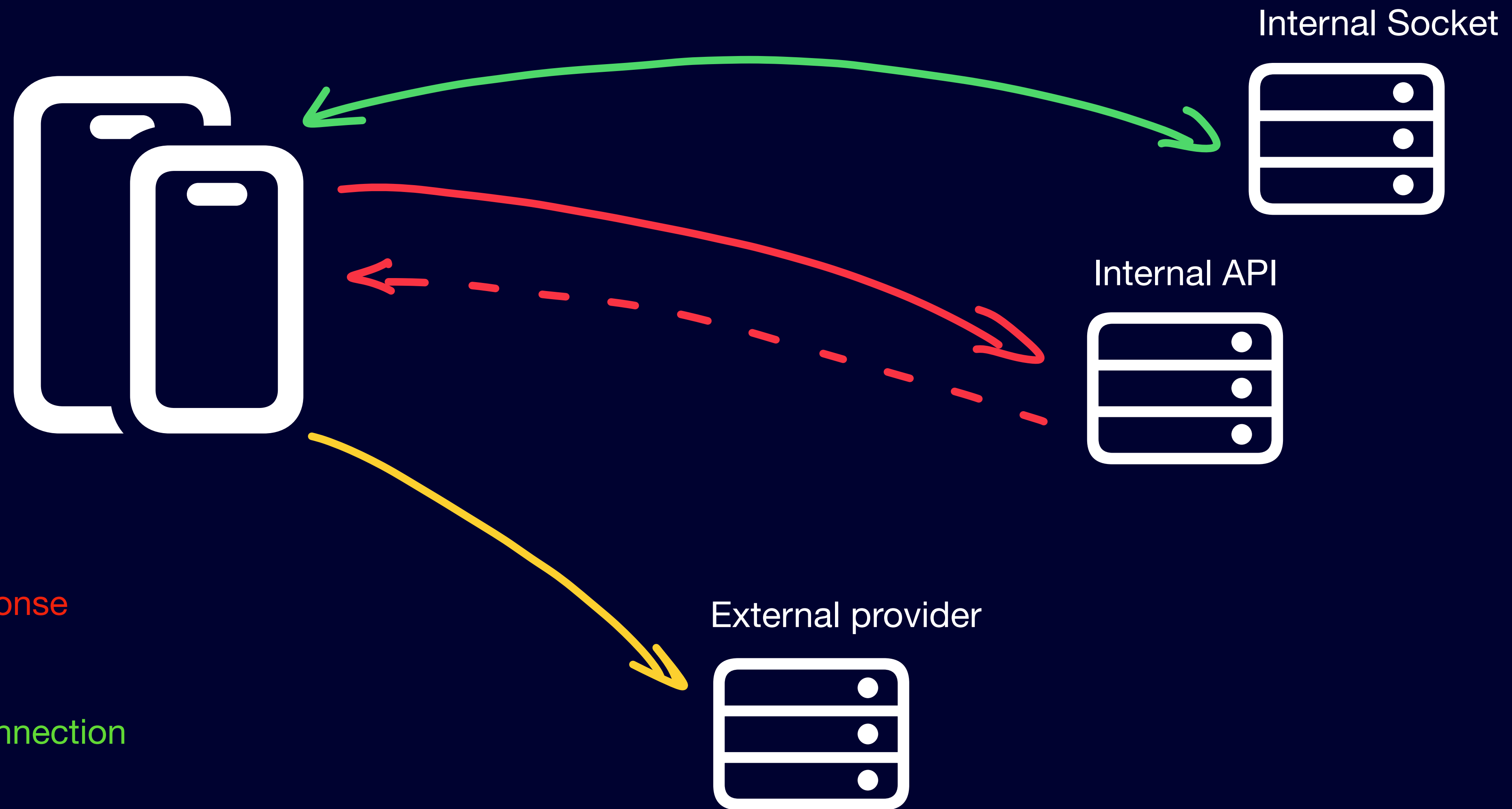
Push / Pull

Bidirectional connection

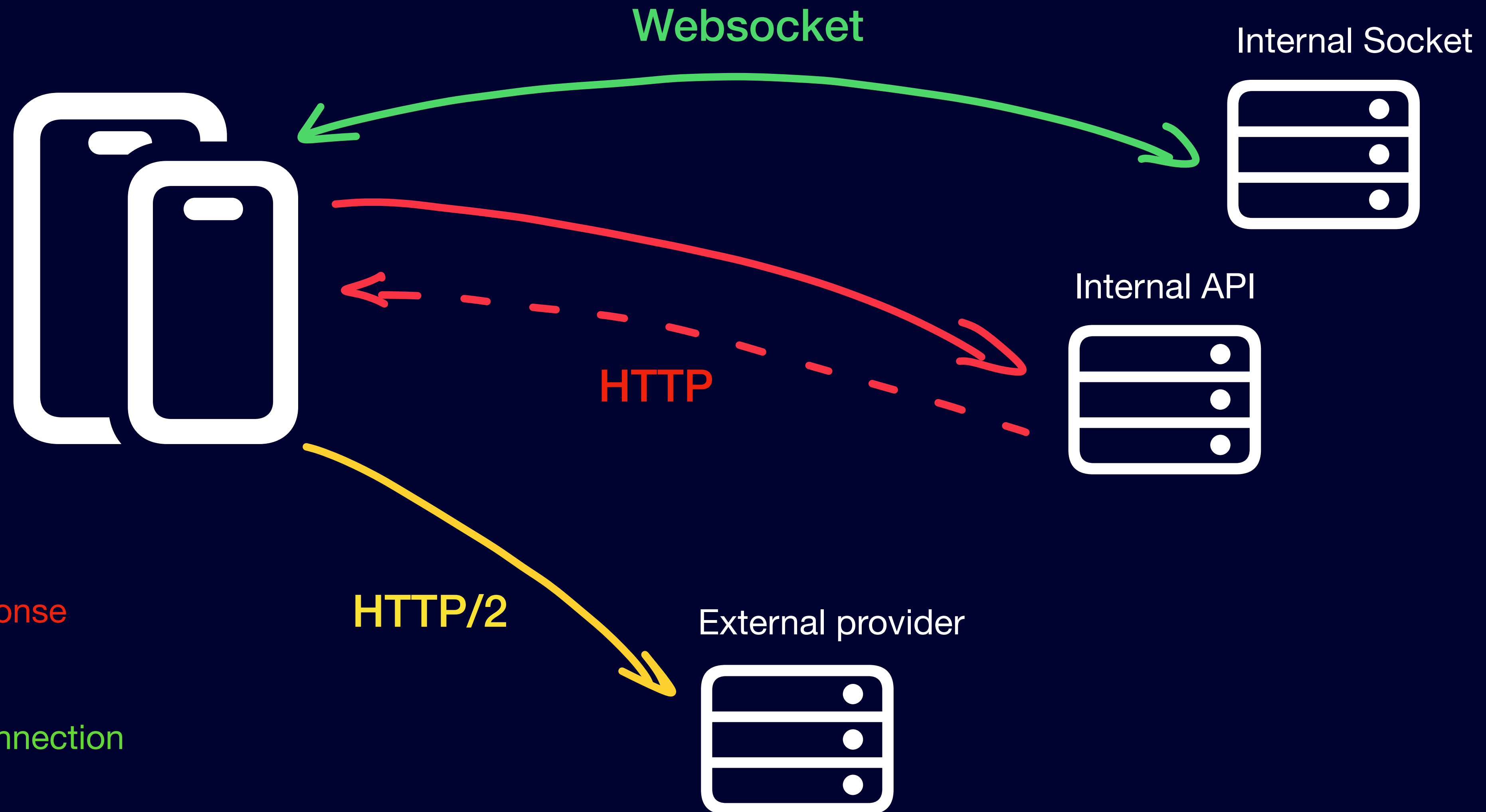
External provider



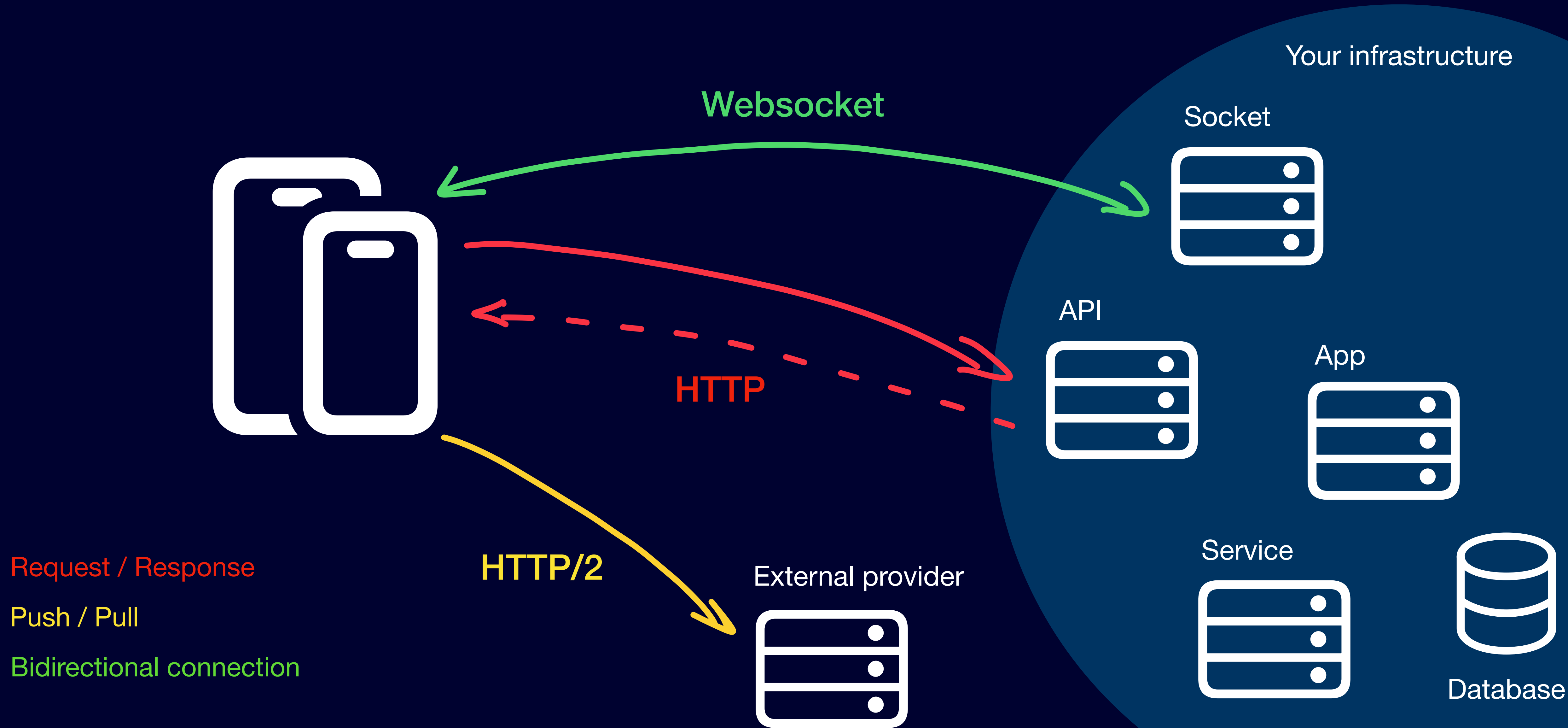
# Communication between your app and servers



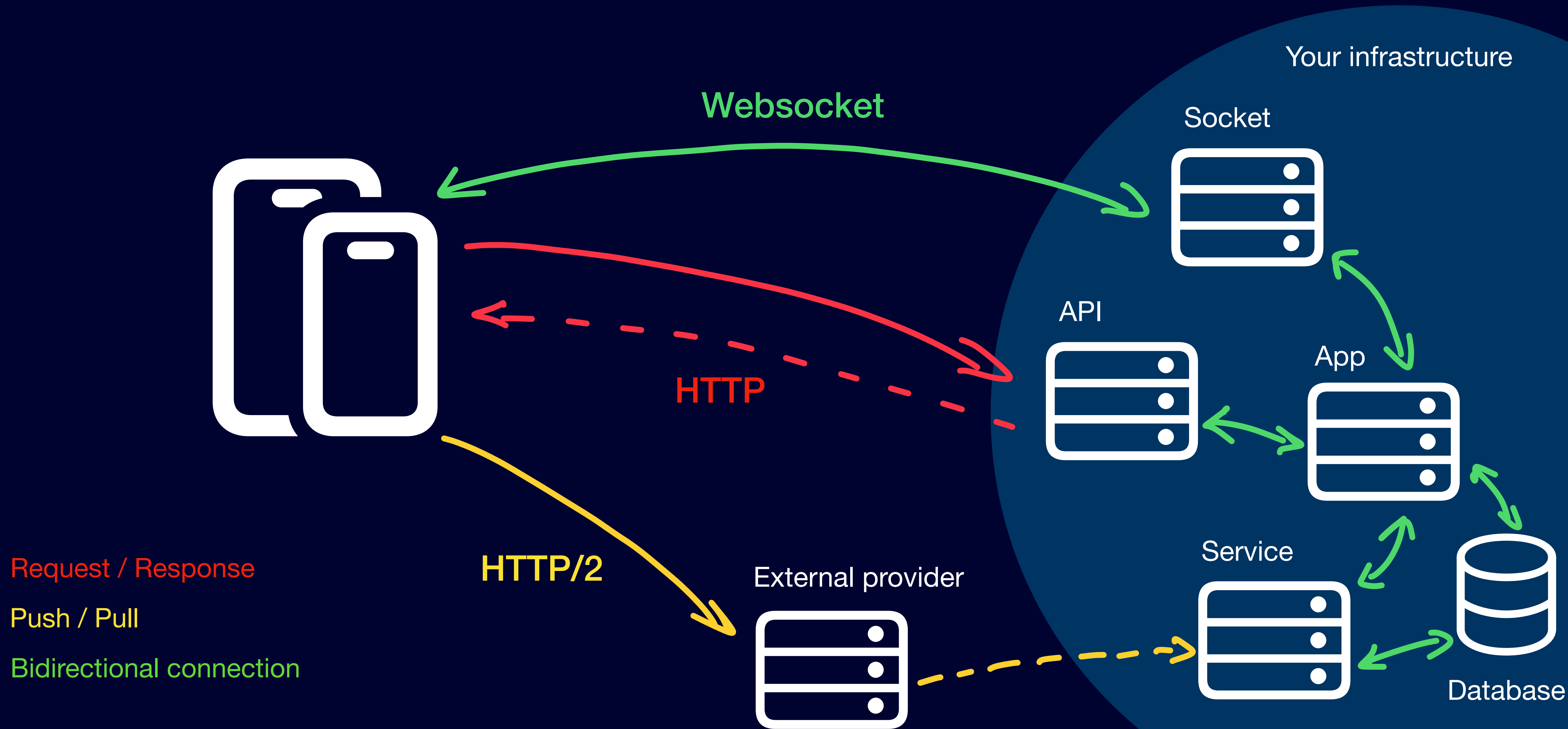
# Communication between your app and servers



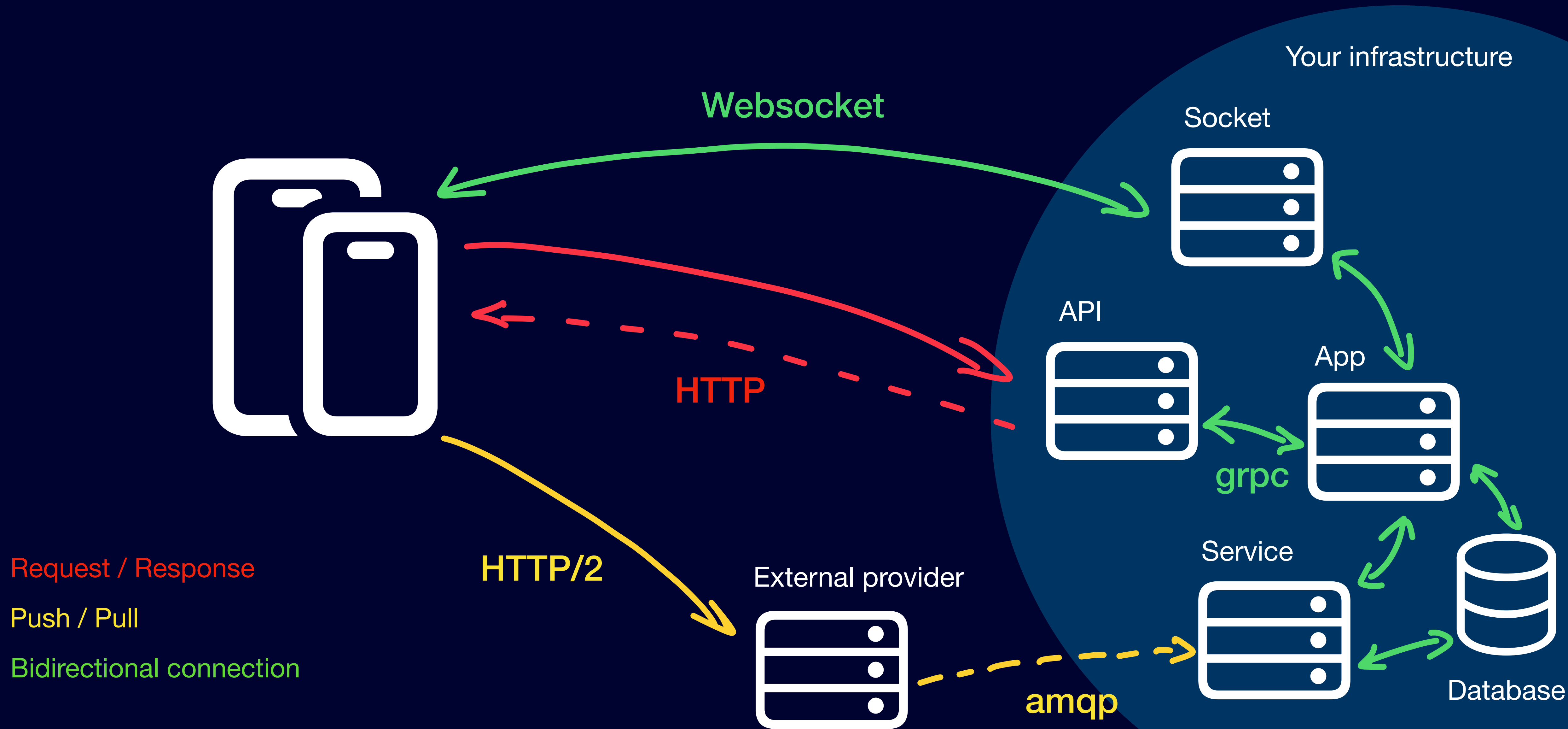
# Communication between your app and the mess



# Communication between your app and the mess



# Communication between your app and the mess



# Different communication standards...

HTTP

Websocket

grpc

HTTP/2

amqp

...

**... all based on two IP protocols ...**

**TCP**

**UDP**



... all based on two IP protocols ...

TCP



UDP

... all based on two IP protocols ...

TCP

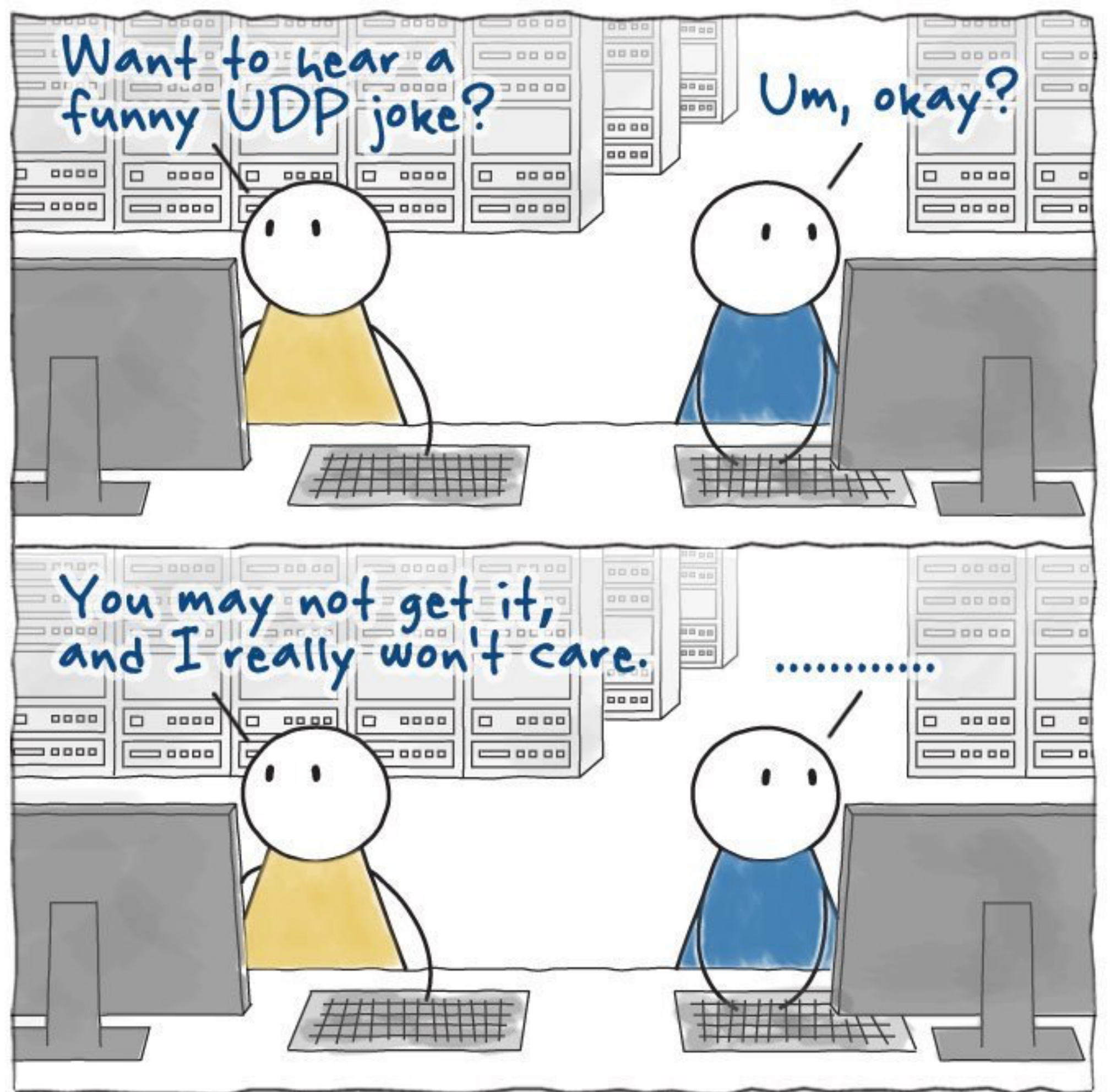


UDP





... all based on two IP protocols ...



www.NeweggBusiness.com

Design: Dana Choi

TCP



UDP



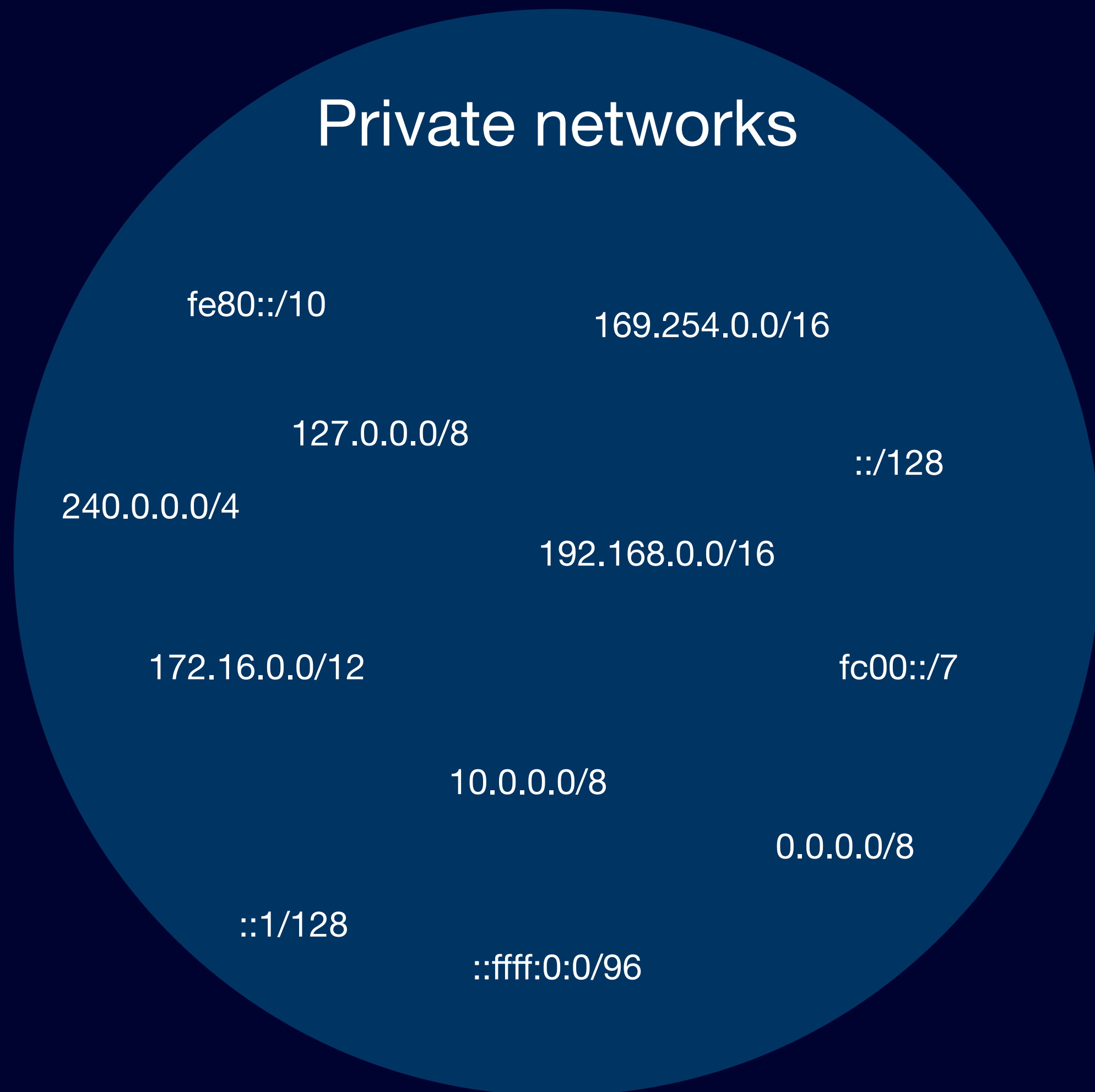
**... over private or public networks**



Private networks

Public network

# ... over private or public networks



Public network



**All communication that go  
through the public network is**

All communication that go  
through the public network is

**public**

**How do you protect against interception?  
alteration?**



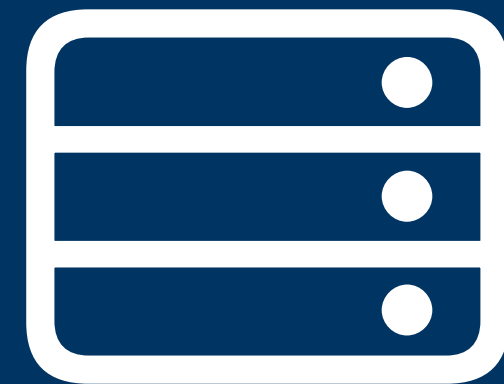
**How do you protect against interception?  
alteration?**

**You don't**

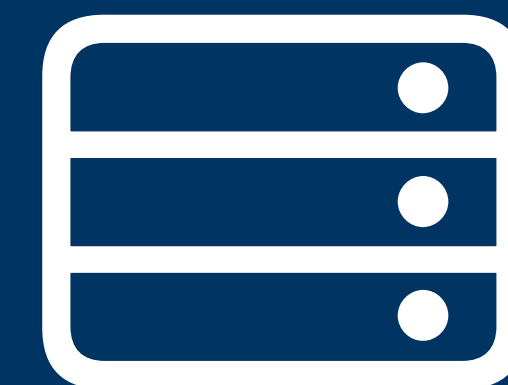
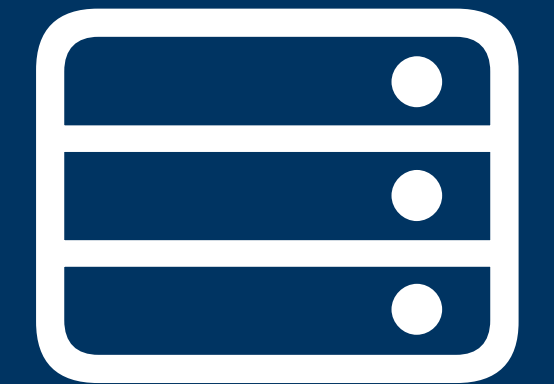
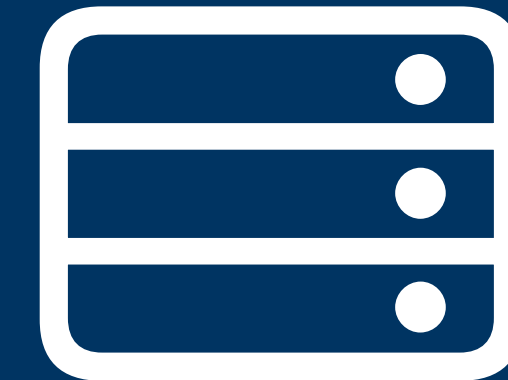
# What can we trust?



External provider



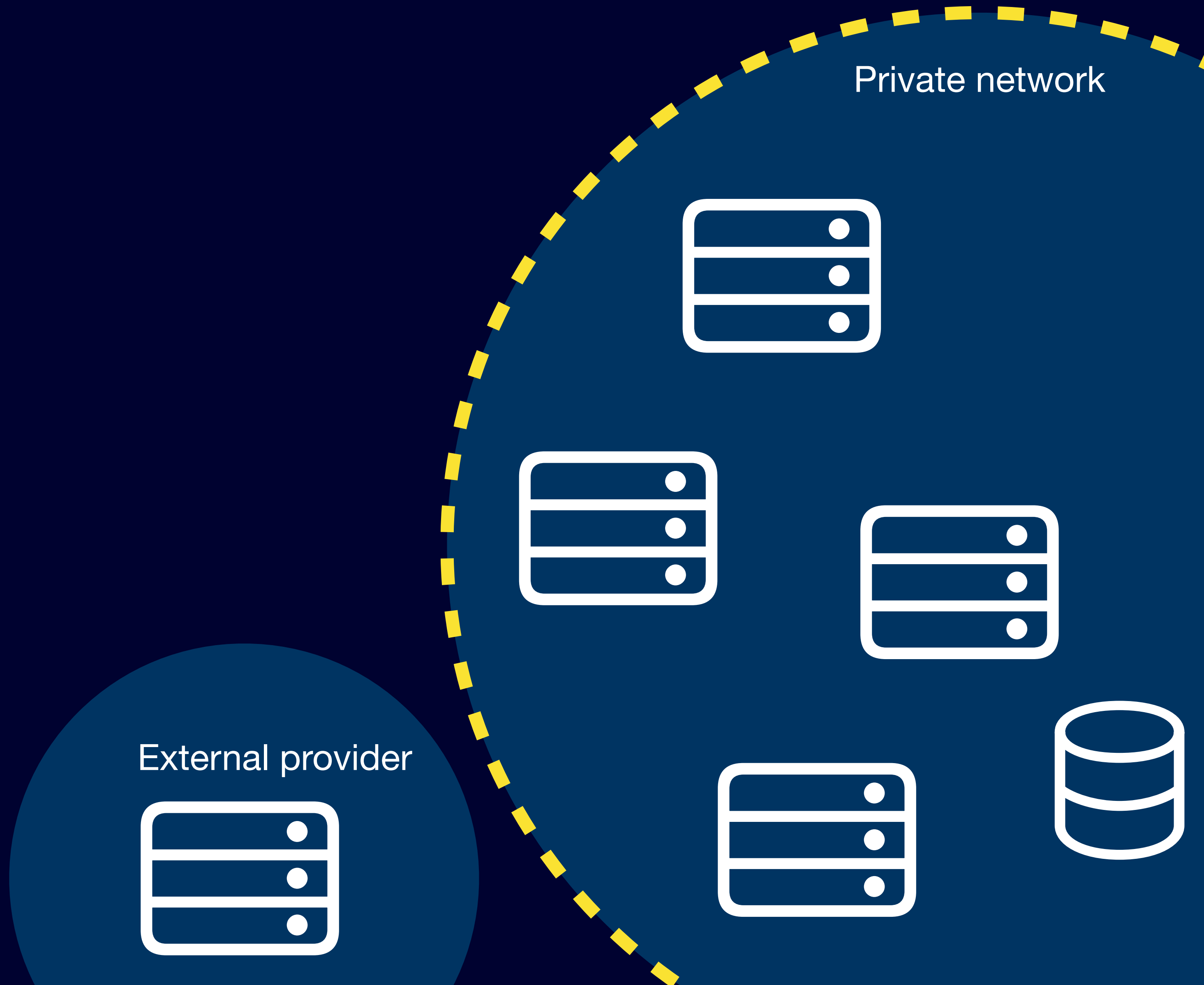
Private network



# What can we trust?



# What can we trust?



Private network

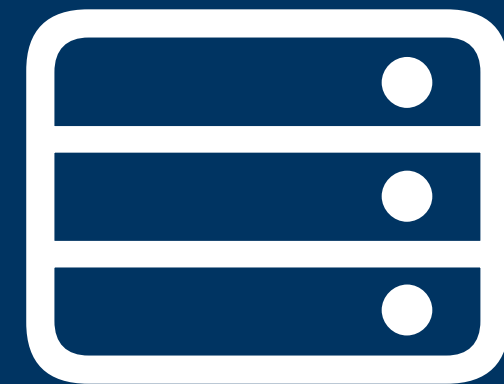
External provider

# What can we trust?

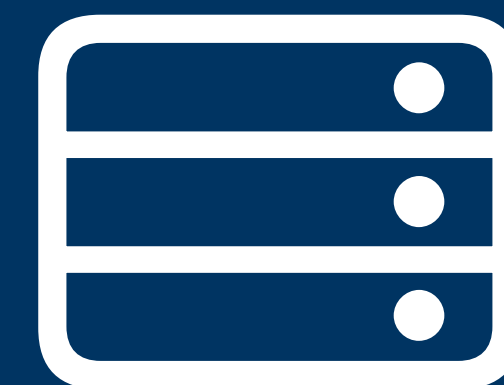
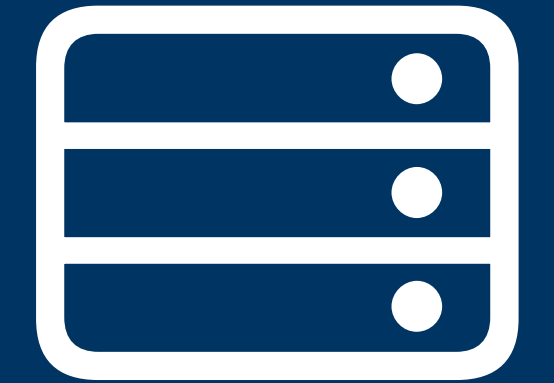
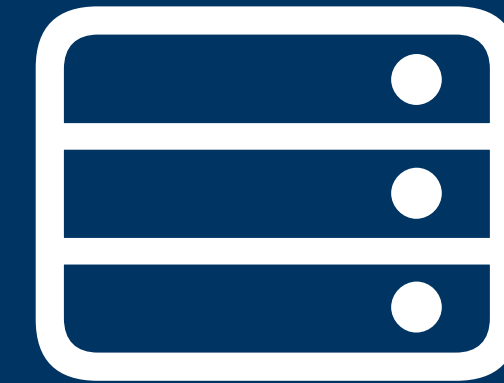
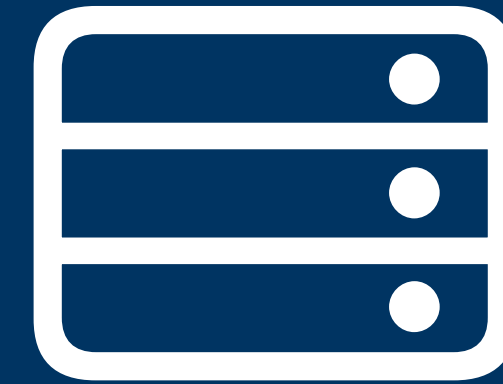


**Danger Zone**

External provider



Private network



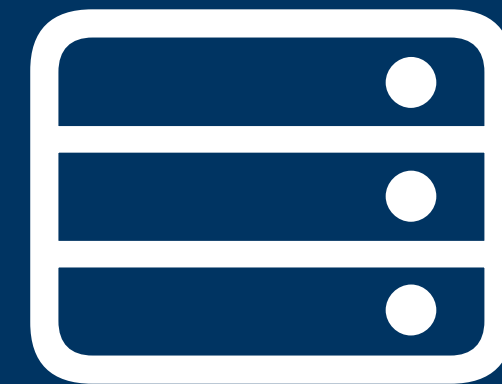
# What can we trust?

Only your servers are  
suitable to make proper  
security checks

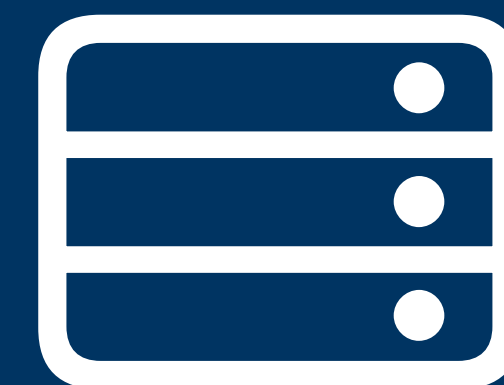
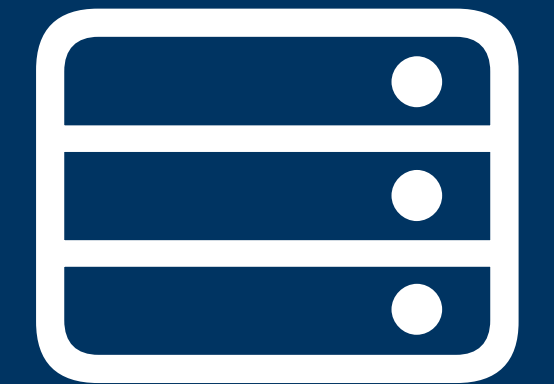
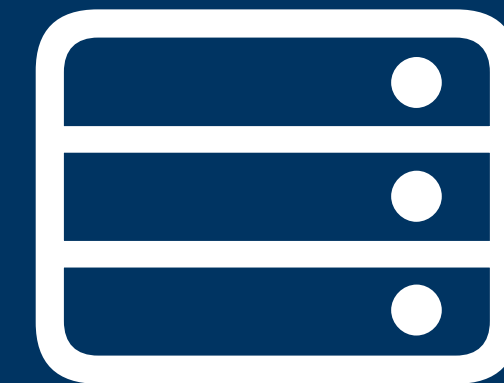
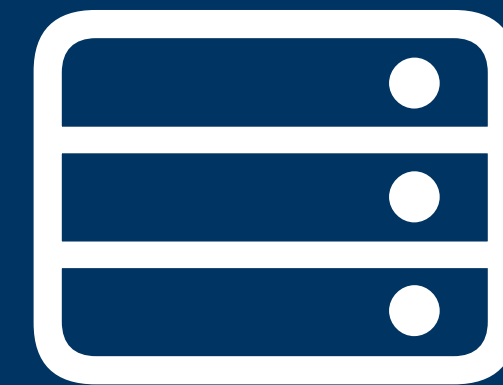


**Danger Zone**

External provider



Private network



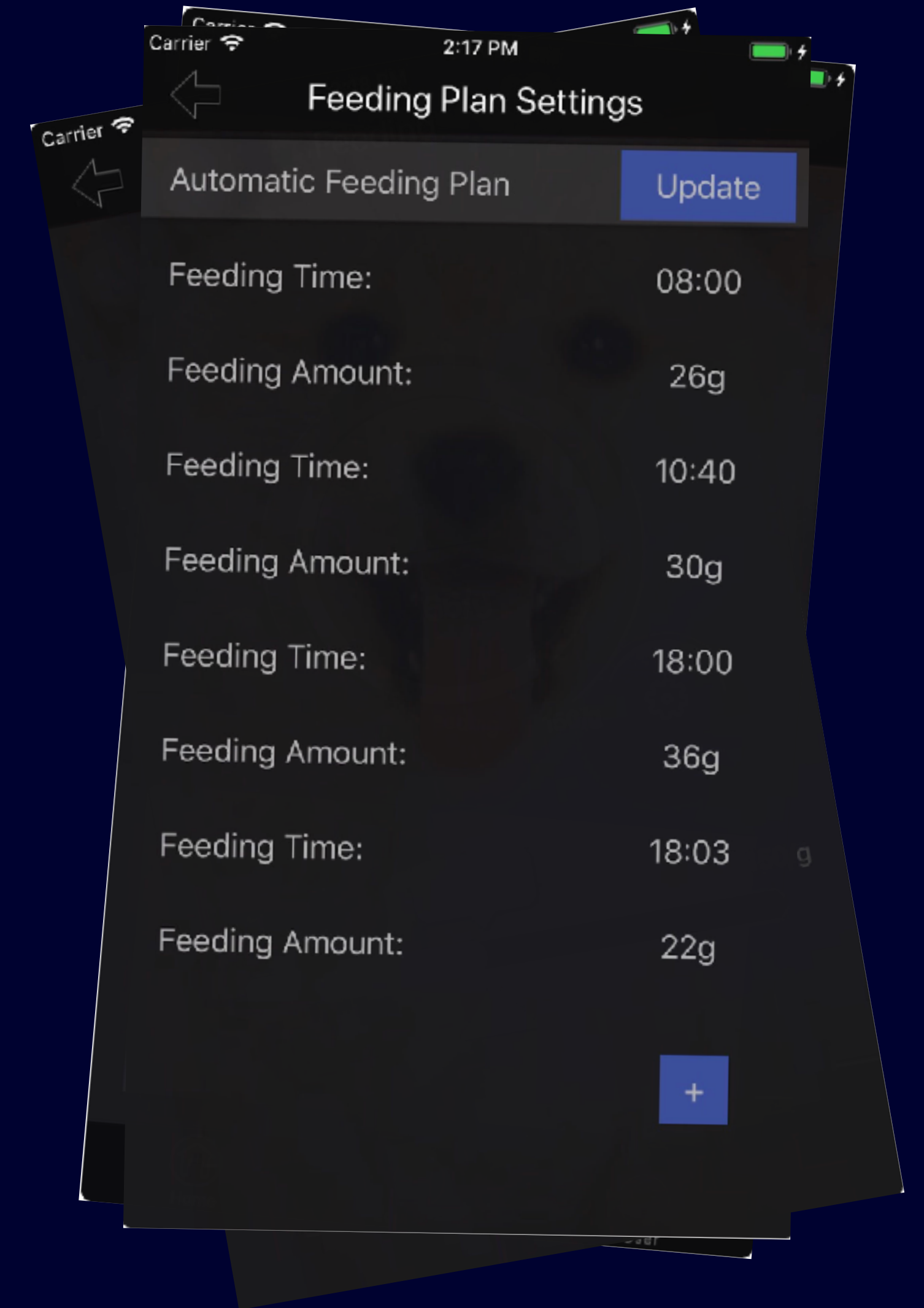
# Interlude #1

# Interlude #1





# Interlude #1





# Reverse-engineering the API

# Reverse-engineering the API



Burp Suite

# Reverse-engineering the API



Burp Suite



Proxyman

# Reverse-engineering the API



Burp Suite



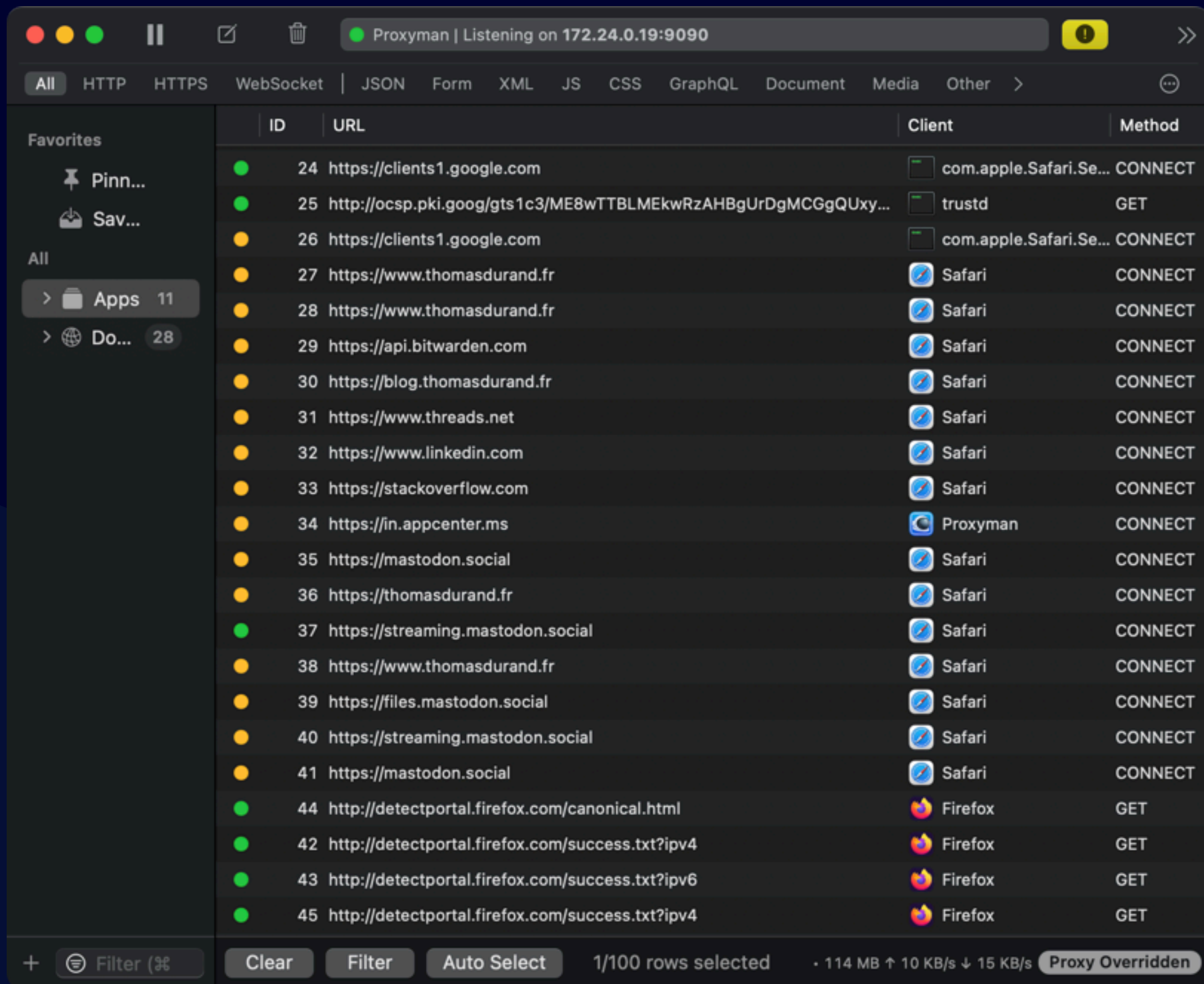
Proxyman

...

# Reverse-engineering the API



# Reverse-engineering the API

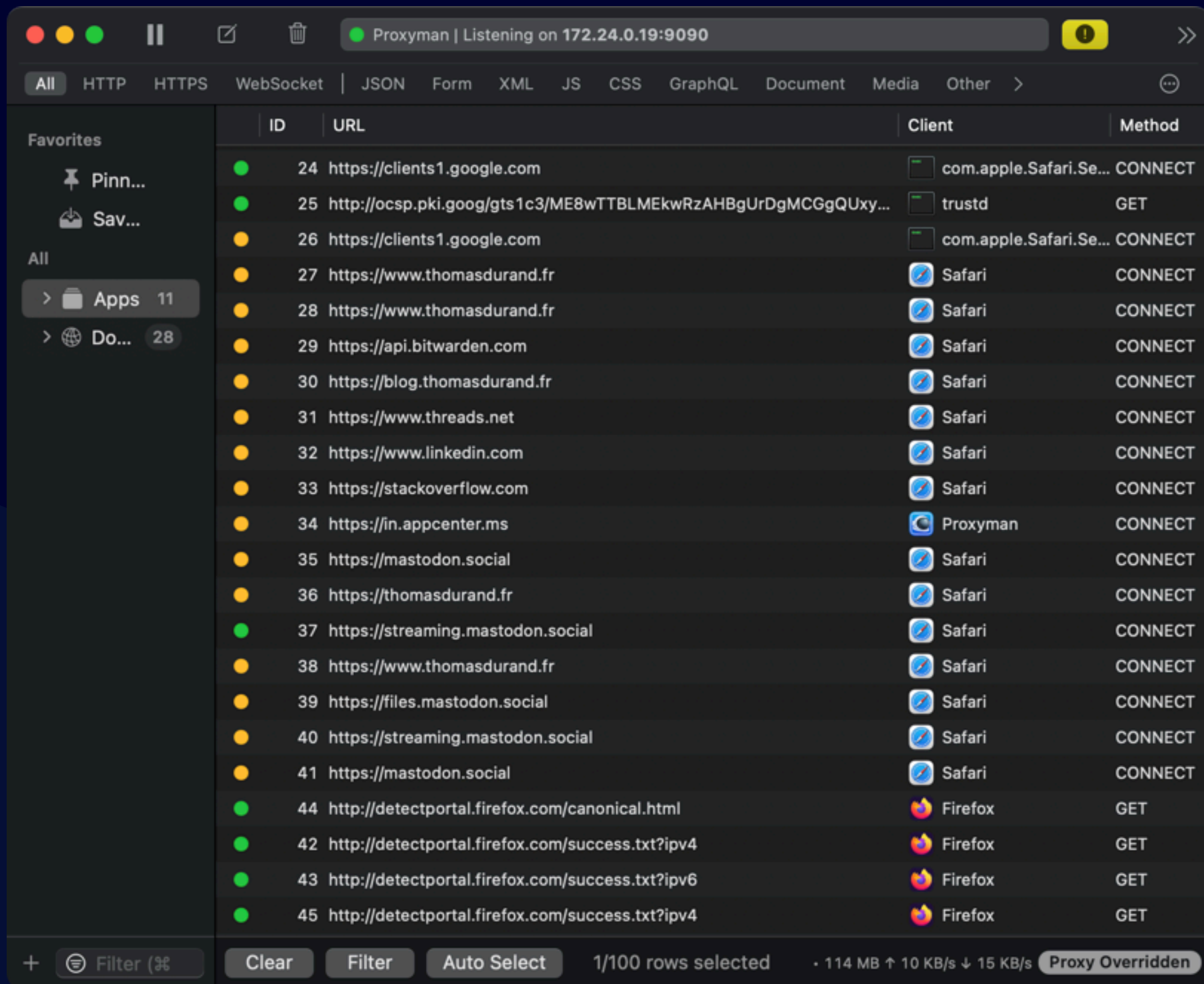


The screenshot shows the Proxyman application interface. The title bar indicates it is listening on 172.24.0.19:9090. The top navigation bar includes filters for All, HTTP, HTTPS, WebSocket, JSON, Form, XML, JS, CSS, GraphQL, Document, Media, and Other. The main area displays a table of intercepted requests with columns for ID, URL, Client, and Method. The status bar at the bottom shows 1/100 rows selected and a proxy override status.

ID	URL	Client	Method
24	https://clients1.google.com	com.apple.Safari.Se...	CONNECT
25	http://ocsp.pki.goog/gts1c3/ME8wTTBLMEkwRzAHBgUrDgMCGgQUxy...	trustd	GET
26	https://clients1.google.com	com.apple.Safari.Se...	CONNECT
27	https://www.thomasdurand.fr	Safari	CONNECT
28	https://www.thomasdurand.fr	Safari	CONNECT
29	https://api.bitwarden.com	Safari	CONNECT
30	https://blog.thomasdurand.fr	Safari	CONNECT
31	https://www.threads.net	Safari	CONNECT
32	https://www.linkedin.com	Safari	CONNECT
33	https://stackoverflow.com	Safari	CONNECT
34	https://in.appcenter.ms	Proxyman	CONNECT
35	https://mastodon.social	Safari	CONNECT
36	https://thomasdurand.fr	Safari	CONNECT
37	https://streaming.mastodon.social	Safari	CONNECT
38	https://www.thomasdurand.fr	Safari	CONNECT
39	https://files.mastodon.social	Safari	CONNECT
40	https://streaming.mastodon.social	Safari	CONNECT
41	https://mastodon.social	Safari	CONNECT
44	http://detectportal.firefox.com/canonical.html	Firefox	GET
42	http://detectportal.firefox.com/success.txt?ipv4	Firefox	GET
43	http://detectportal.firefox.com/success.txt?ipv6	Firefox	GET
45	http://detectportal.firefox.com/success.txt?ipv4	Firefox	GET



# Reverse-engineering the API

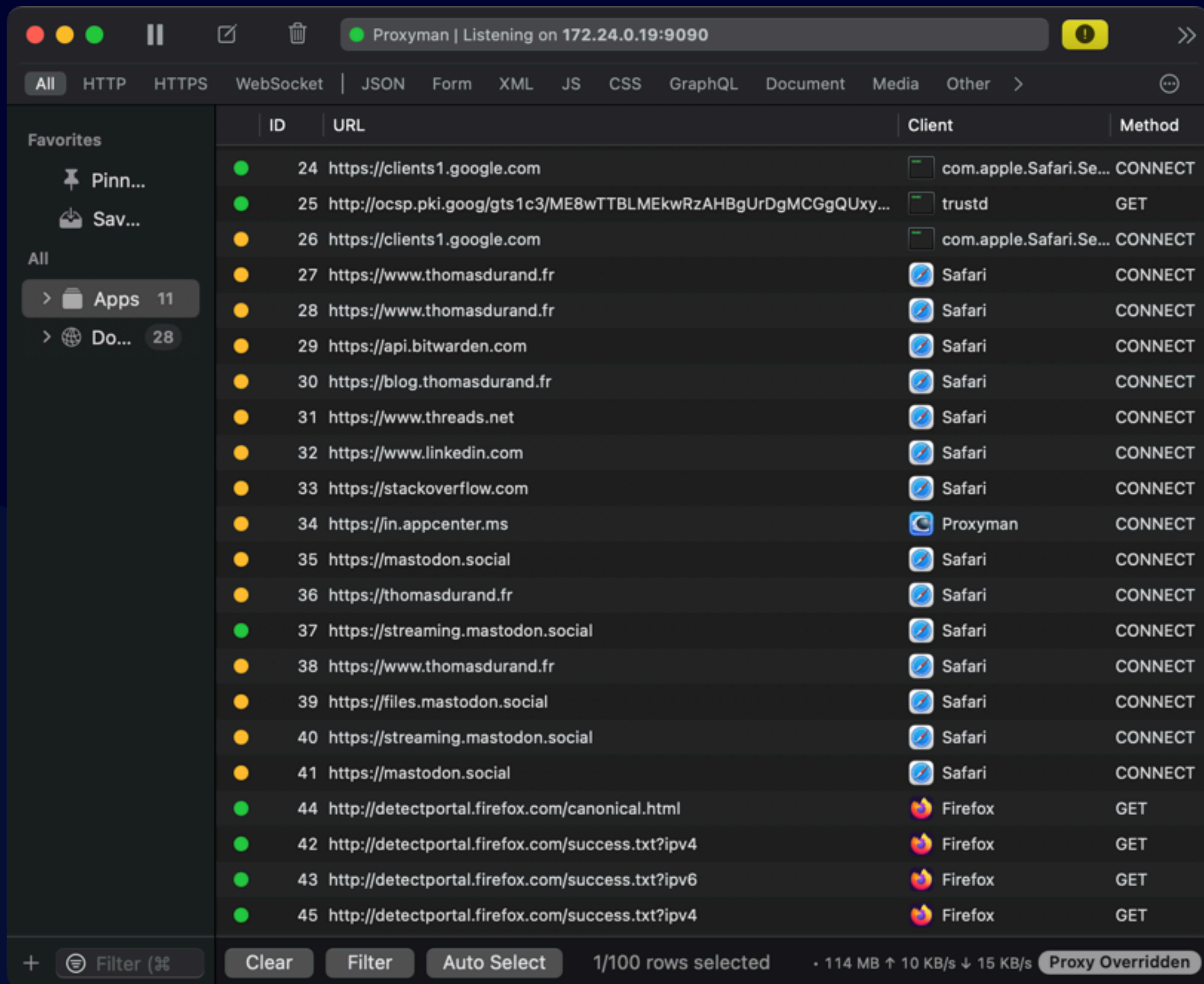


The screenshot shows the Proxyman application interface. The top bar indicates it is listening on 172.24.0.19:9090. Below the navigation tabs, a table lists intercepted requests. The table has columns for ID, URL, Client, and Method. The status of each request is indicated by a colored dot in the ID column.

ID	URL	Client	Method
24	https://clients1.google.com	com.apple.Safari.Se...	CONNECT
25	http://ocsp.pki.goog/gts1c3/ME8wTTBLMEkwRzAHBgUrDgMCGgQUxy...	trustd	GET
26	https://clients1.google.com	com.apple.Safari.Se...	CONNECT
27	https://www.thomasdurand.fr	Safari	CONNECT
28	https://www.thomasdurand.fr	Safari	CONNECT
29	https://api.bitwarden.com	Safari	CONNECT
30	https://blog.thomasdurand.fr	Safari	CONNECT
31	https://www.threads.net	Safari	CONNECT
32	https://www.linkedin.com	Safari	CONNECT
33	https://stackoverflow.com	Safari	CONNECT
34	https://in.appcenter.ms	Proxyman	CONNECT
35	https://mastodon.social	Safari	CONNECT
36	https://thomasdurand.fr	Safari	CONNECT
37	https://streaming.mastodon.social	Safari	CONNECT
38	https://www.thomasdurand.fr	Safari	CONNECT
39	https://files.mastodon.social	Safari	CONNECT
40	https://streaming.mastodon.social	Safari	CONNECT
41	https://mastodon.social	Safari	CONNECT
44	http://detectportal.firefox.com/canonical.html	Firefox	GET
42	http://detectportal.firefox.com/success.txt?ipv4	Firefox	GET
43	http://detectportal.firefox.com/success.txt?ipv6	Firefox	GET
45	http://detectportal.firefox.com/success.txt?ipv4	Firefox	GET

`http://fr.dev.alnpet.com`  
`http://us1.dev.alnpet.com`

# Reverse-engineering the API



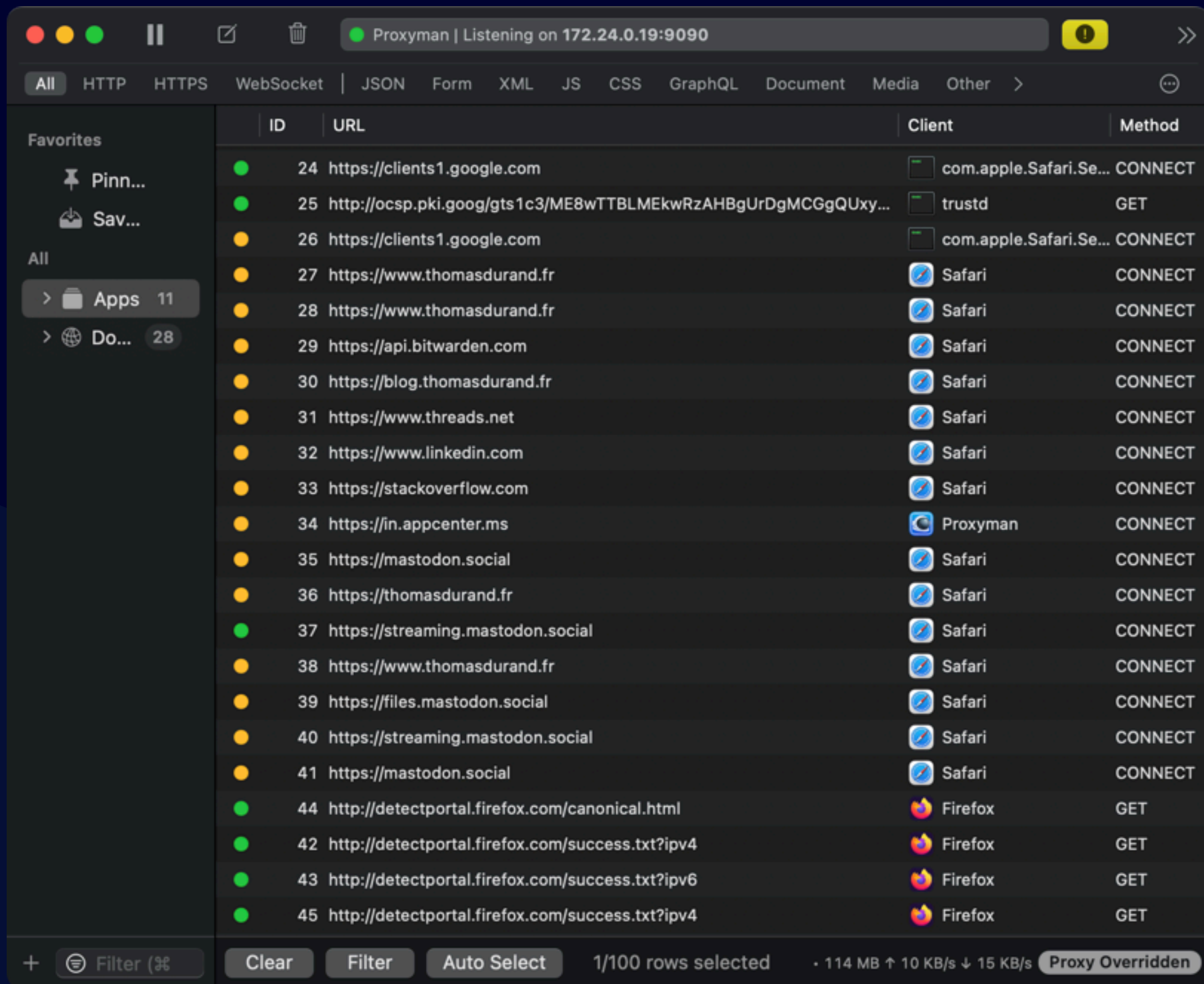
The screenshot shows the Proxyman application interface. The main window displays a list of intercepted network requests. The table has columns for ID, URL, Client, and Method. The status bar at the bottom indicates that 1/100 rows are selected and that the proxy is overridden.

ID	URL	Client	Method
24	https://clients1.google.com	com.apple.Safari.Se...	CONNECT
25	http://ocsp.pki.goog/gts1c3/ME8wTTBLMEkwRzAHBgUrDgMCGgQUxy...	trustd	GET
26	https://clients1.google.com	com.apple.Safari.Se...	CONNECT
27	https://www.thomasdurand.fr	Safari	CONNECT
28	https://www.thomasdurand.fr	Safari	CONNECT
29	https://api.bitwarden.com	Safari	CONNECT
30	https://blog.thomasdurand.fr	Safari	CONNECT
31	https://www.threads.net	Safari	CONNECT
32	https://www.linkedin.com	Safari	CONNECT
33	https://stackoverflow.com	Safari	CONNECT
34	https://in.appcenter.ms	Proxyman	CONNECT
35	https://mastodon.social	Safari	CONNECT
36	https://thomasdurand.fr	Safari	CONNECT
37	https://streaming.mastodon.social	Safari	CONNECT
38	https://www.thomasdurand.fr	Safari	CONNECT
39	https://files.mastodon.social	Safari	CONNECT
40	https://streaming.mastodon.social	Safari	CONNECT
41	https://mastodon.social	Safari	CONNECT
44	http://detectportal.firefox.com/canonical.html	Firefox	GET
42	http://detectportal.firefox.com/success.txt?ipv4	Firefox	GET
43	http://detectportal.firefox.com/success.txt?ipv6	Firefox	GET
45	http://detectportal.firefox.com/success.txt?ipv4	Firefox	GET

http://fr.dev.a1npet.com  
http://us1.dev.a1npet.com



# Reverse-engineering the API



The screenshot shows the Proxyman application interface. The main window displays a list of intercepted network requests. The table has columns for ID, URL, Client, and Method. The status bar at the bottom indicates '1/100 rows selected' and 'Proxy Overridden'.

ID	URL	Client	Method
24	https://clients1.google.com	com.apple.Safari.Se...	CONNECT
25	http://ocsp.pki.goog/gts1c3/ME8wTTBLMEkwRzAHBgUrDgMCGgQUxy...	trustd	GET
26	https://clients1.google.com	com.apple.Safari.Se...	CONNECT
27	https://www.thomasdurand.fr	Safari	CONNECT
28	https://www.thomasdurand.fr	Safari	CONNECT
29	https://api.bitwarden.com	Safari	CONNECT
30	https://blog.thomasdurand.fr	Safari	CONNECT
31	https://www.threads.net	Safari	CONNECT
32	https://www.linkedin.com	Safari	CONNECT
33	https://stackoverflow.com	Safari	CONNECT
34	https://in.appcenter.ms	Proxyman	CONNECT
35	https://mastodon.social	Safari	CONNECT
36	https://thomasdurand.fr	Safari	CONNECT
37	https://streaming.mastodon.social	Safari	CONNECT
38	https://www.thomasdurand.fr	Safari	CONNECT
39	https://files.mastodon.social	Safari	CONNECT
40	https://streaming.mastodon.social	Safari	CONNECT
41	https://mastodon.social	Safari	CONNECT
44	http://detectportal.firefox.com/canonical.html	Firefox	GET
42	http://detectportal.firefox.com/success.txt?ipv4	Firefox	GET
43	http://detectportal.firefox.com/success.txt?ipv6	Firefox	GET
45	http://detectportal.firefox.com/success.txt?ipv4	Firefox	GET

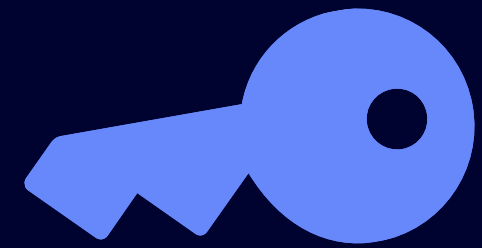
http://fr.dev.alnpet.com  
http://us1.dev.alnpet.com

**How do you protect against data interception?  
alteration?**

**How do you protect against data interception?  
alteration?**

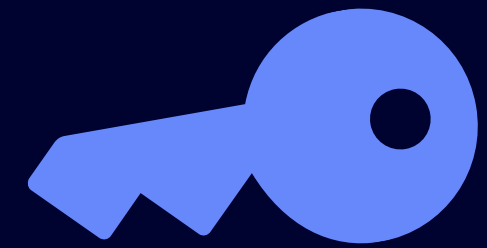
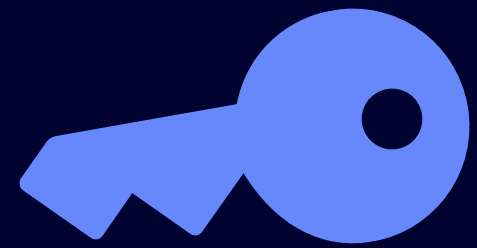
**Cryptography**

# Cryptography



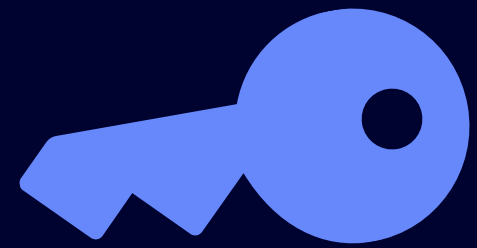
# Cryptography

## Shared Key

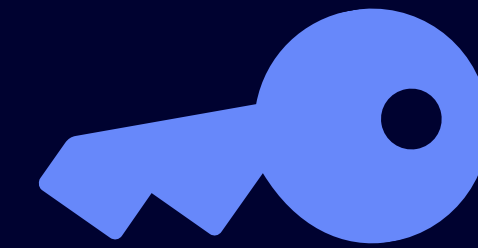


# Cryptography

## Shared Key



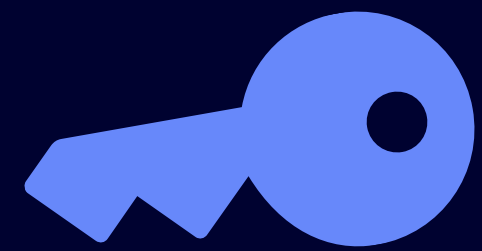
Hello World!



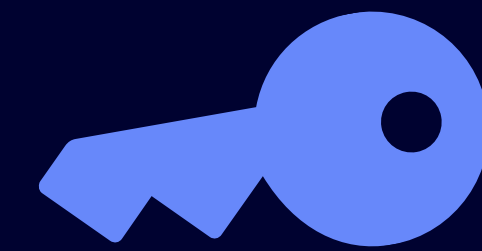


# Cryptography

## Shared Key

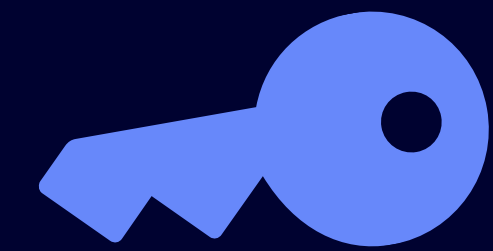
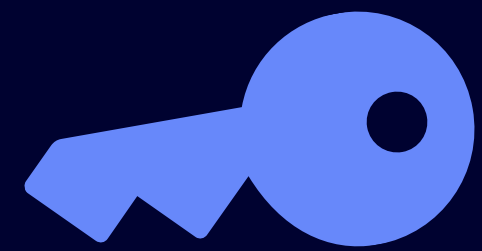


OiZpTPcVEGV7PWG5vO94vw



# Cryptography

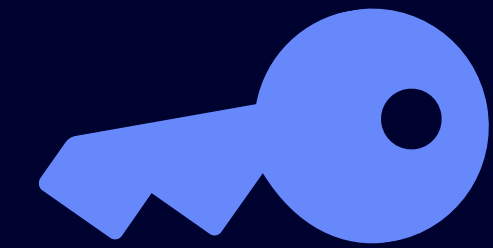
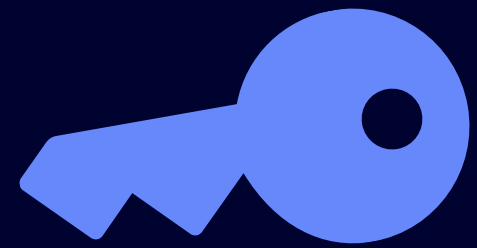
## Shared Key



OiZpTPcVEGV7PWG5vO94vw

# Cryptography

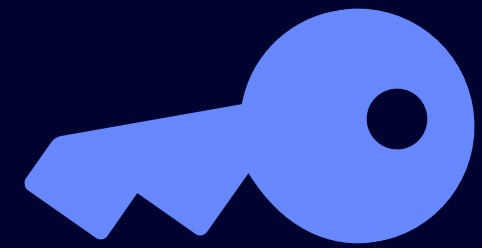
## Shared Key



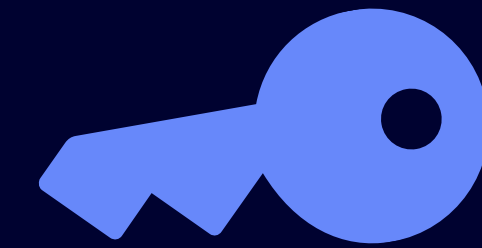
Hello World!

# Cryptography

Shared Key

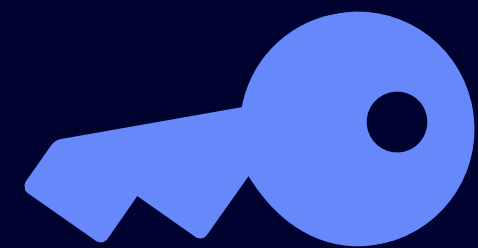


Symmetric-Key encryption

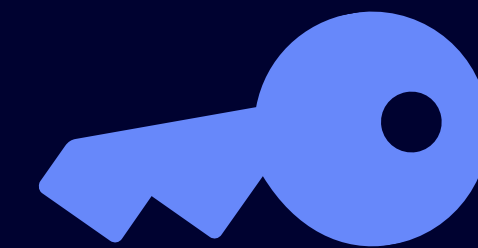


# Cryptography

## Shared Key



Symmetric-Key encryption



**How do you share the key?**

# Cryptography



# Cryptography



# Cryptography



Public key





# Cryptography



Public key



Private key

# Cryptography

Shared public key



# Cryptography

Shared public key



Hello World!



# Cryptography

Shared public key



TW9iaWxpc0luTW9iaWxIMjAyNA



# Cryptography

Shared public key



TW9iaWxpc0luTW9iaWxlMjAyNA

# Cryptography

Shared public key



Hello World!

# Cryptography

Shared public key





# Cryptography

Shared public key



Thanks!

# Cryptography

Shared public key



IVBkYBapZ9jipbatHLO7Hw

# Cryptography

Shared public key



IVBkYBapZ9jipbatHLO7Hw



# Cryptography

Shared public key



Thanks!



# Cryptography

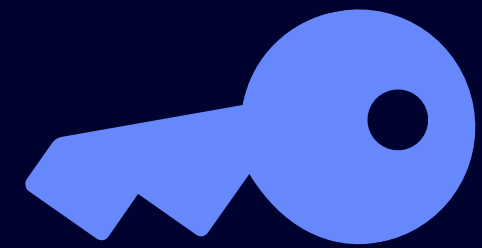
Shared public key



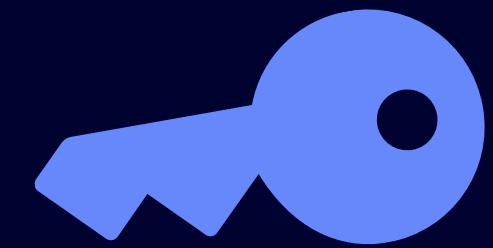
Asymmetric-Key encryption



# Cryptography



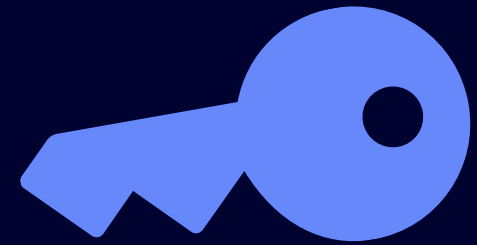
Symmetric-Key encryption



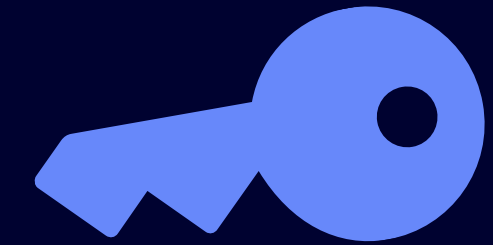
Asymmetric-Key encryption



# Cryptography



Symmetric-Key encryption

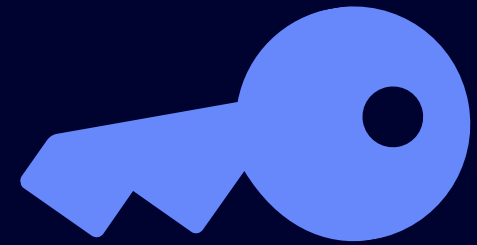


Asymmetric-Key encryption

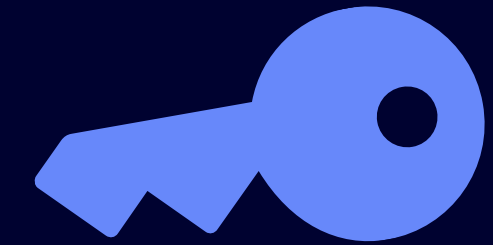


# TLS

# Cryptography



Symmetric-Key encryption



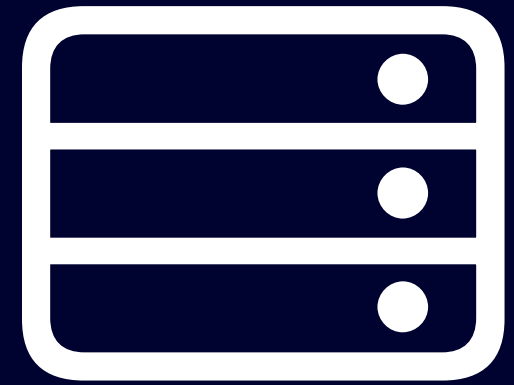
Asymmetric-Key encryption



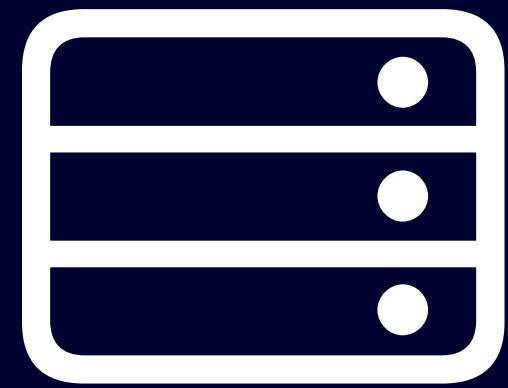
# Transport Layer Security



# Transport Layer Security



# Transport Layer Security

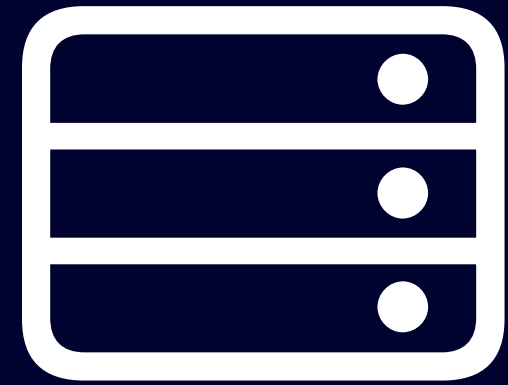


Key



Certificate

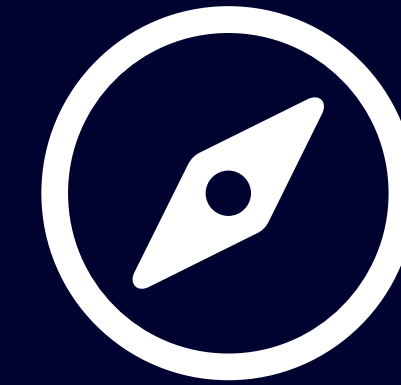
# Transport Layer Security



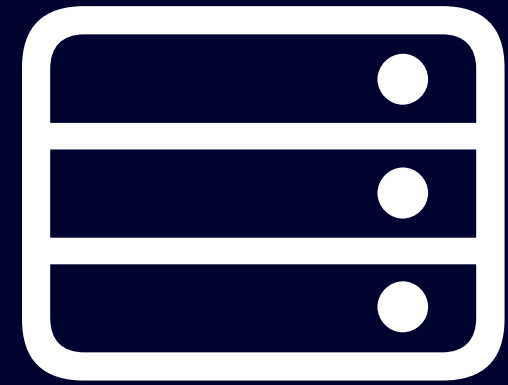
Key



Certificate



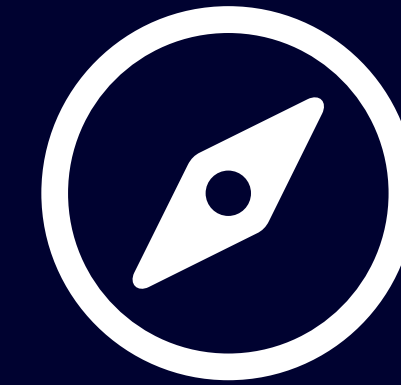
# Transport Layer Security



Key

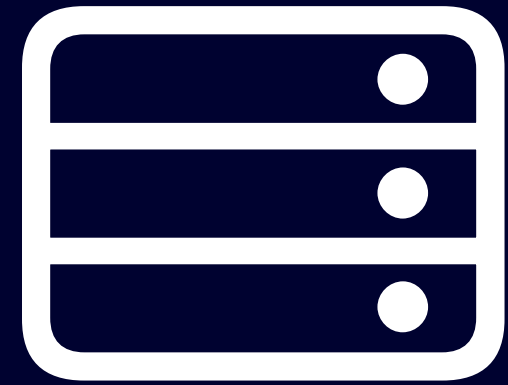


Certificate



Certificate

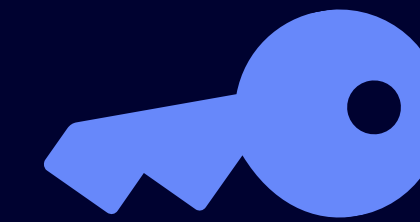
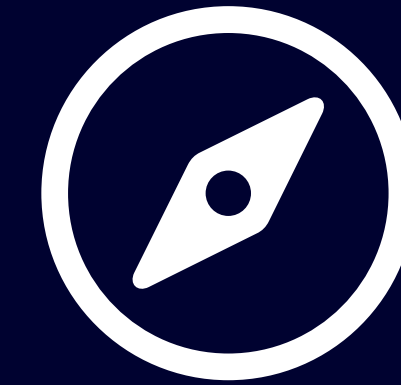
# Transport Layer Security



Key



Certificate

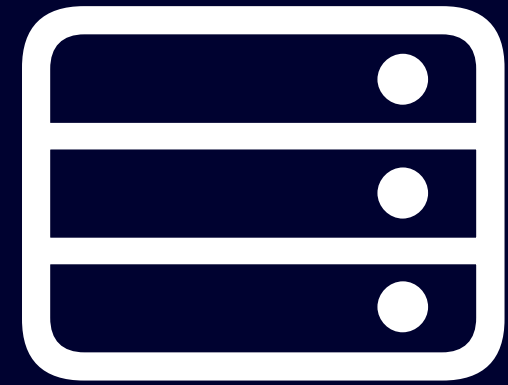


Client Key



Certificate

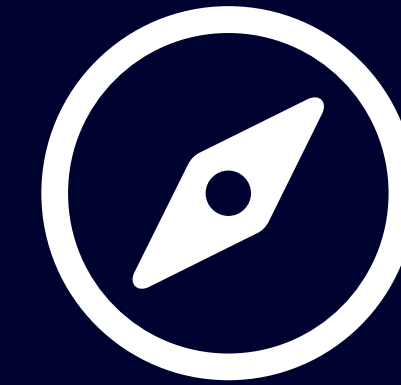
# Transport Layer Security



Key



Certificate

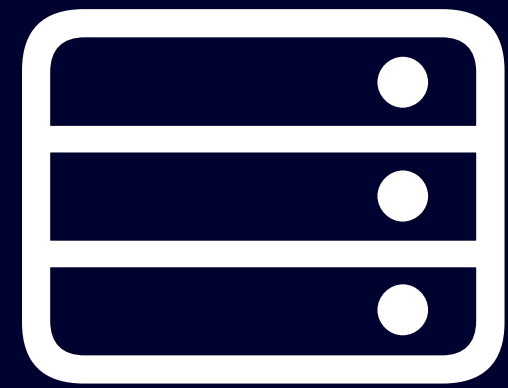


Secret



Certificate

# Transport Layer Security

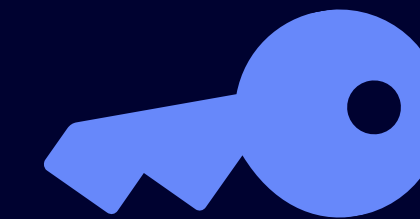
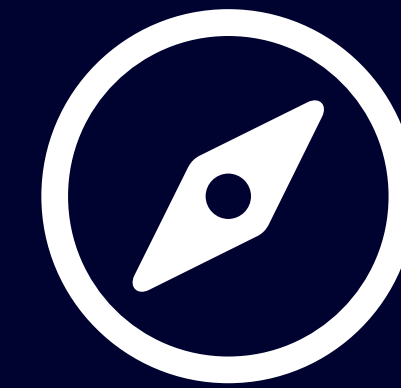


Key

Secret



Certificate

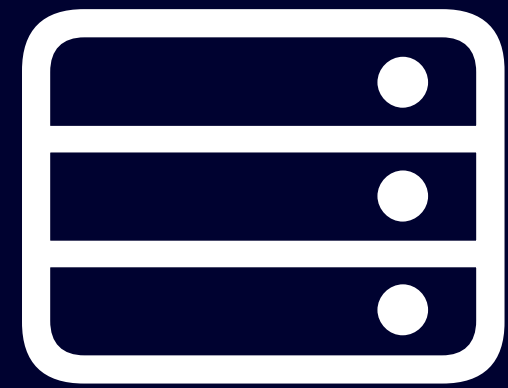


Client Key



Certificate

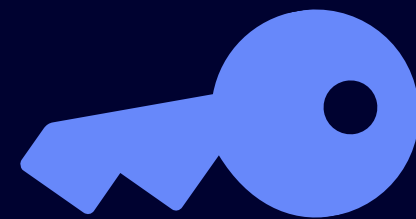
# Transport Layer Security



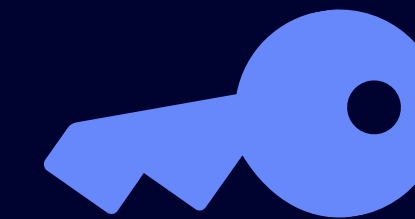
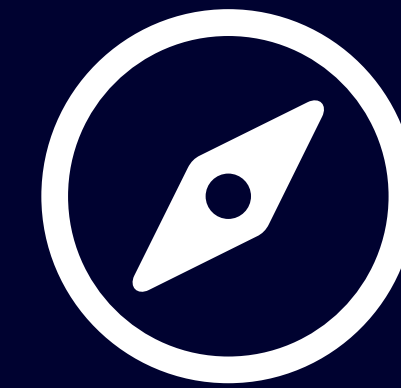
Key



Certificate



Client Key



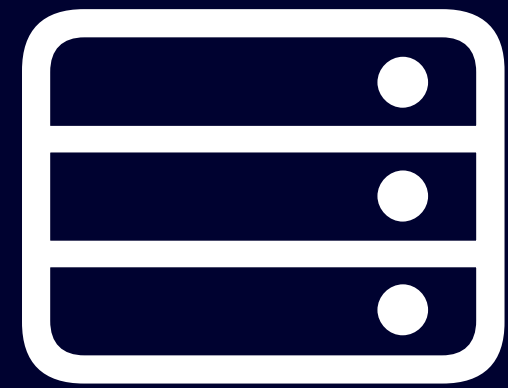
Client Key



Certificate



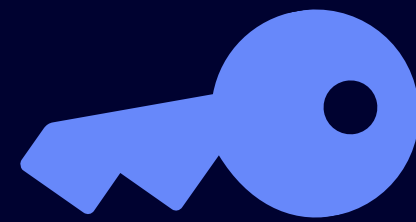
# Transport Layer Security



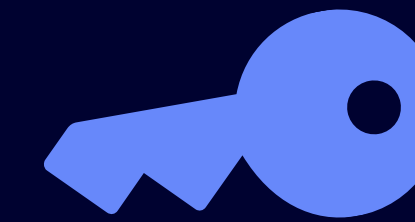
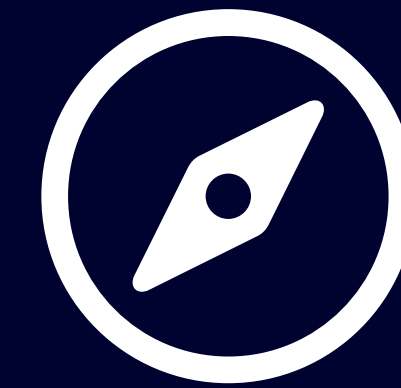
Key



Certificate



Client Key

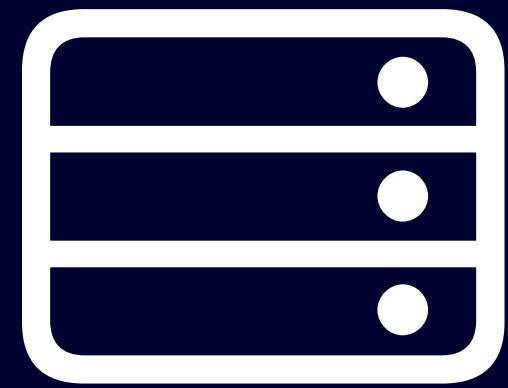


Client Key



Certificate

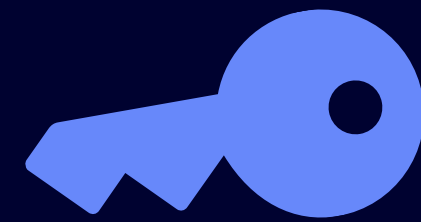
# Transport Layer Security



Key

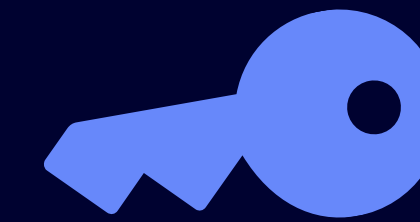
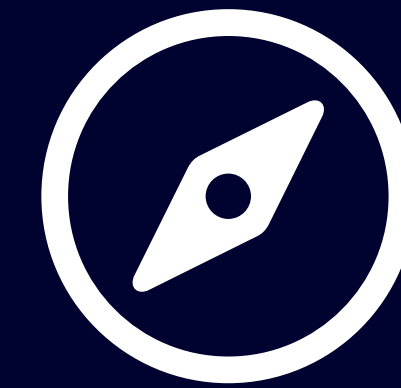


Certificate



Client Key

## Handshake



Client Key



Certificate

# Transport Layer Security

# Transport Layer Security

- Ensure the **confidentiality**, and **integrity** of data in transit

# Transport Layer Security

- Ensure the **confidentiality**, and **integrity** of data in transit
  - Data can't be read by 

# Transport Layer Security

- Ensure the **confidentiality**, and **integrity** of data in transit
  - Data can't be read by 🦴
  - Data can't be altered by 🦴

# Transport Layer Security

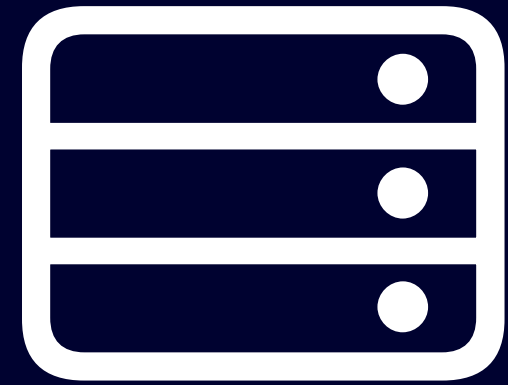
- Ensure the **confidentiality**, and **integrity** of data in transit
  - Data can't be read by 🦴
  - Data can't be altered by 🦴
- Certificate comes from the target server

# Transport Layer Security

- Ensure the **confidentiality**, and **integrity** of data in transit
  - Data can't be read by 🦴
  - Data can't be altered by 🦴
- Certificate comes from the target server
  - Authenticity of the server is **NOT ensured** (yet)



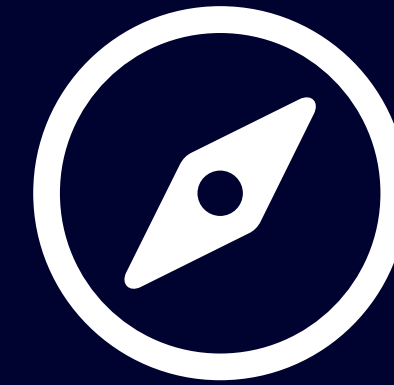
# Transport Layer Security



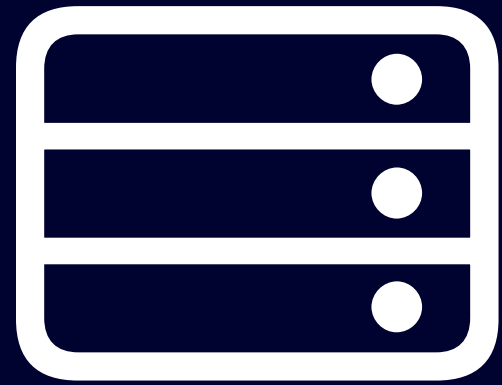
Key



Certificate



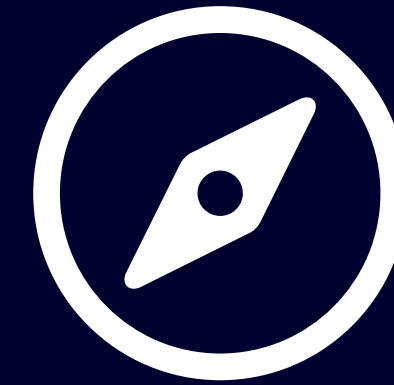
# Transport Layer Security



Key



Certificate



Certificate

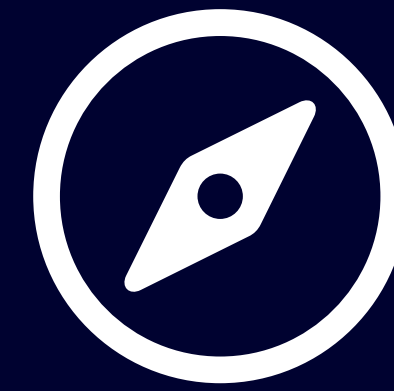
# Transport Layer Security



Key



Certificate



Certificate

Server is not guaranteed to be trustworthy

# Transport Layer Security



Key



Attacker Key



Certificate



Impersonated Certificate

# Transport Layer Security



Key



Certificate



Impersonated Certificate



Impersonated Certificate

# Transport Layer Security



Key

**Man in the middle attack**



Certificate



Impersonated Certificate



Impersonated Certificate

# Transport Layer Security

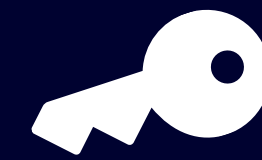


Certificate

# Transport Layer Security



Certificate



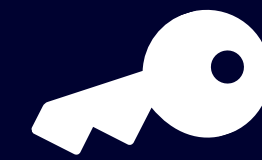
Public key



# Transport Layer Security



Certificate



Public key

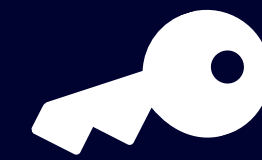


Domain name

# Transport Layer Security



Certificate



Public key



Domain name

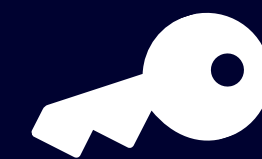


Expiration

# Transport Layer Security



Certificate



Public key



Domain name



Expiration



Certificate Authority

# Transport Layer Security

 Certificate Authority



Key



Certificate

# Transport Layer Security

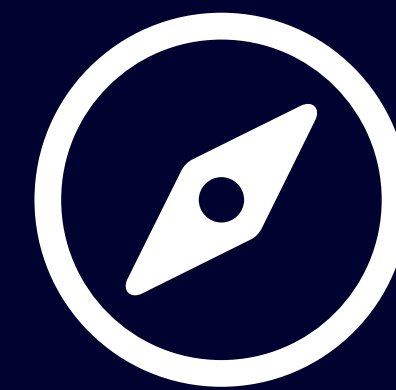
 Certificate Authority



Key



Certificate



Certificate

# Transport Layer Security

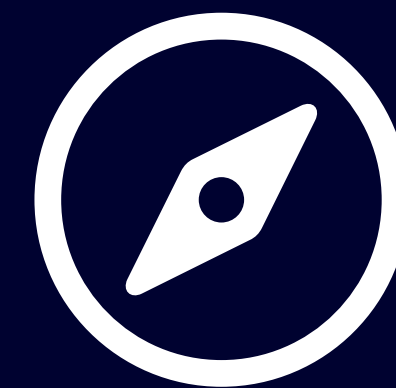
 Certificate Authority



Key



Certificate



Certificate

Provided by system update

# Transport Layer Security

 Certificate Authority

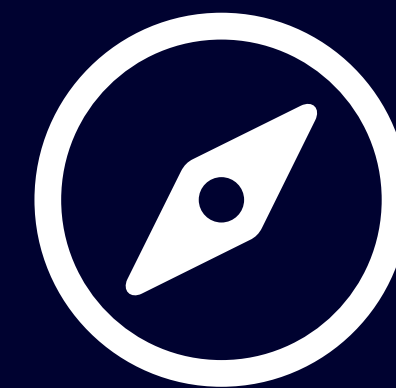


Key



Certificate

Valid for decades



Certificate



# Transport Layer Security

 Certificate Authority



Key

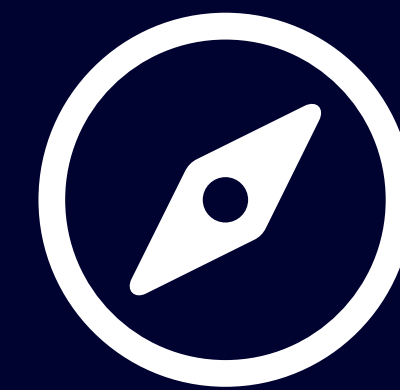


Key

(Intermediate)



Certificate



Certificate

Valid for years



# Transport Layer Security

 Certificate Authority



Key

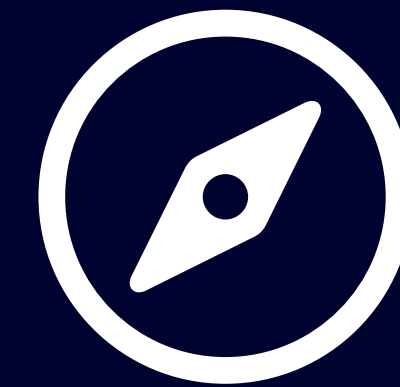


Key

(Intermediate)



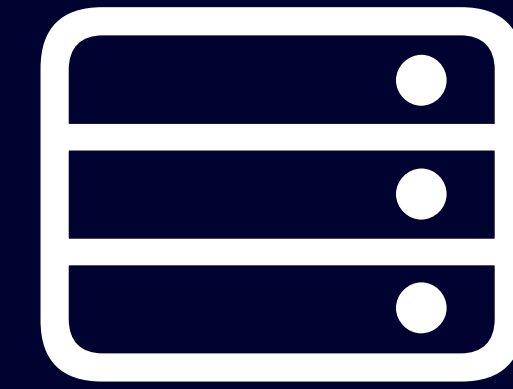
Certificate



Certificate

Valid for years

# Transport Layer Security



 Certificate Authority



Key



Key

(Intermediate)

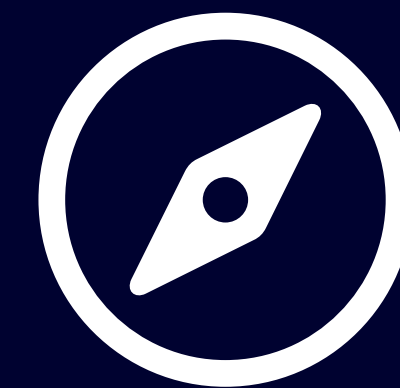


Key

(server)



Certificate



Certificate

Valid for months

# Transport Layer Security



## Certificate Authority



Key

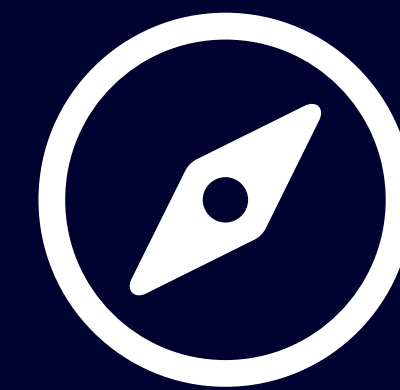


Key

(Intermediate)



Certificate



Certificate

# Transport Layer Security

 Certificate Authority



Key

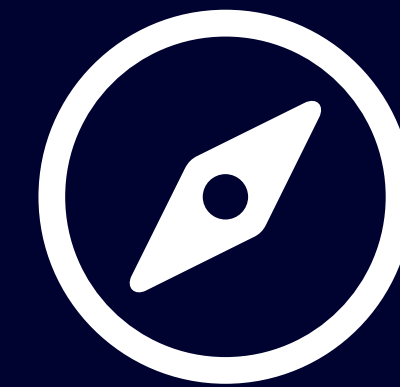
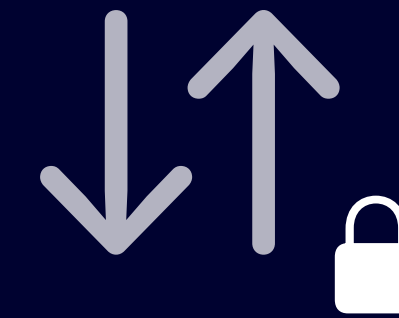
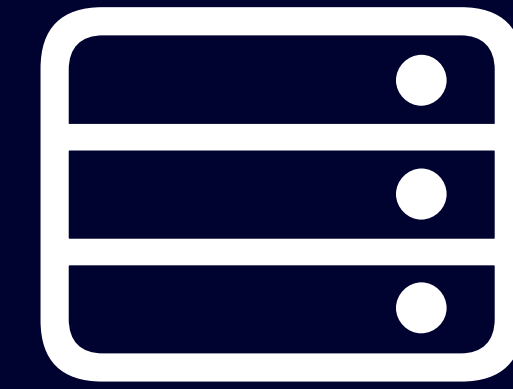


Key

(Intermediate)



Certificate



Certificate

# Transport Layer Security

Is man-in-the-middle attack still possible?

# Transport Layer Security

Is man-in-the-middle attack still possible?

- Server private key leak (unlikely)

# Transport Layer Security

Is man-in-the-middle attack still possible?

- Server private key leak (unlikely)
  - Make sure your server is safely guarded

# Transport Layer Security

Is man-in-the-middle attack still possible?

- Server private key leak (unlikely)
  - Make sure your server is safely guarded
- CA private key leak (**very** unlikely)



# Transport Layer Security

Is man-in-the-middle attack still possible?

- Server private key leak (unlikely)
  - Make sure your server is safely guarded
- CA private key leak (**very** unlikely)
- Security flaw in TLS (likely to almost impossible)

# Transport Layer Security

Is man-in-the-middle attack still possible?

- Server private key leak (unlikely)
  - Make sure your server is safely guarded
- CA private key leak (**very** unlikely)
- Security flaw in TLS (likely to almost impossible)
  - Depends on both server & client version

# Transport Layer Security

Is man-in-the-middle attack still possible?

- Server private key leak (unlikely)
  - Make sure your server is safely guarded
- CA private key leak (**very** unlikely)
- Security flaw in TLS (likely to almost impossible)
  - Depends on both server & client version
- Attacker Certificate Authority trusted by the client machine (**likely**)

# Transport Layer Security

## Is man-in-the-middle attack still possible?

- Server private key leak (unlikely)
  - Make sure your server is safely guarded
- CA private key leak (**very** unlikely)
- Security flaw in TLS (likely to almost impossible)
  - Depends on both server & client version
- Attacker Certificate Authority trusted by the client machine (**likely**)
  - Your customer devices are not to be trusted!

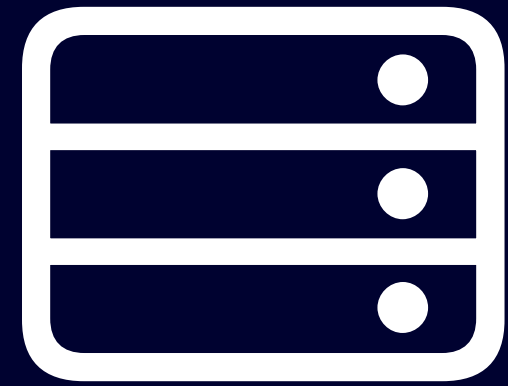
Your users devices are in the  
**danger zone**

# Prevent MitM attacks

## Certificate Pinning

# Prevent MitM attacks

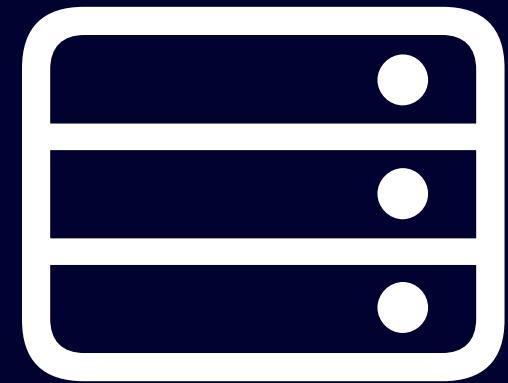
## Certificate Pinning



The server provide  
his certificate

# Prevent MitM attacks

## Certificate Pinning



The server provide  
his certificate

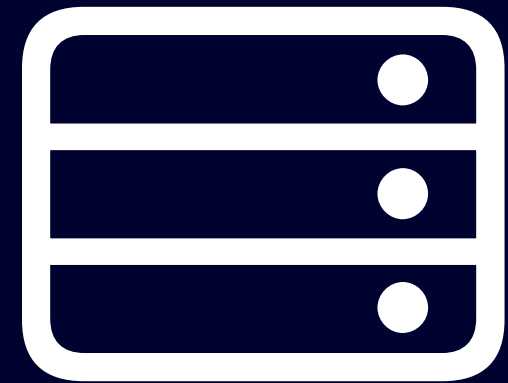


The system checks  
standard TLS



# Prevent MitM attacks

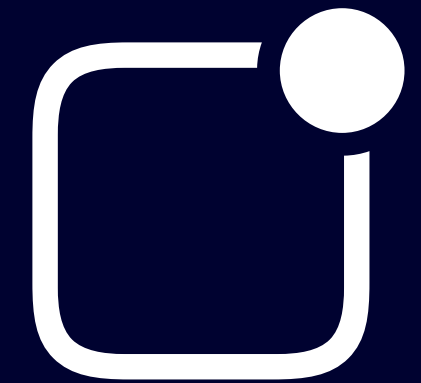
## Certificate Pinning



The server provide  
his certificate



The system checks  
standard TLS

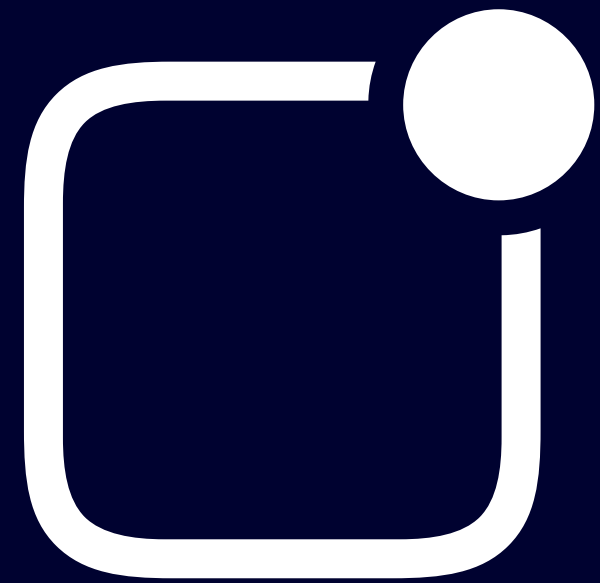


Your app cross-check  
if the certificate is known

# Prevent MitM attacks

## Certificate Pinning

- Your app cross-check if the certificate is known



# Prevent MitM attacks

## Certificate Pinning

- Your app cross-check if the certificate is known



- Your app should ship **with** the certificates you trust

# Prevent MitM attacks

## Certificate Pinning

- Your app cross-check if the certificate is known



- Your app should ship **with** the certificates you trust
- Certificates have expiration date → frequent updates

# Prevent MitM attacks

## Certificate Pinning

- Your app cross-check if the certificate is known



- Your app should ship **with** the certificates you trust
- Certificates have expiration date → frequent updates

- Your app should warn the user for new updates

# Prevent MitM attacks

## Certificate Pinning

- Your app cross-check if the certificate is known
- Your app should ship **with** the certificates you trust
- Certificates have expiration date → frequent updates
- Your app should warn the user for new updates
- Prevent attacker to add/change certificates



# Prevent MitM attacks

## Certificate Pinning



- Your app cross-check if the certificate is known
- Your app should ship **with** the certificates you trust
- Certificates have expiration date → frequent updates
- Your app should warn the user for new updates
- Prevent attacker to add/change certificates
- Your app should prevent the device from being rooted

# Prevent MitM attacks

## Certificate Pinning





# Prevent MitM attacks

Certificate Pinning

# Prevent MitM attacks

Certificate Pinning

Should you?

# Prevent MitM attacks

## Certificate Pinning

Should you?

Is the data you transport is **sensible enough** to justify the troubles of certificate pinning?



# Identification & Authentication

# Identification & Authentication

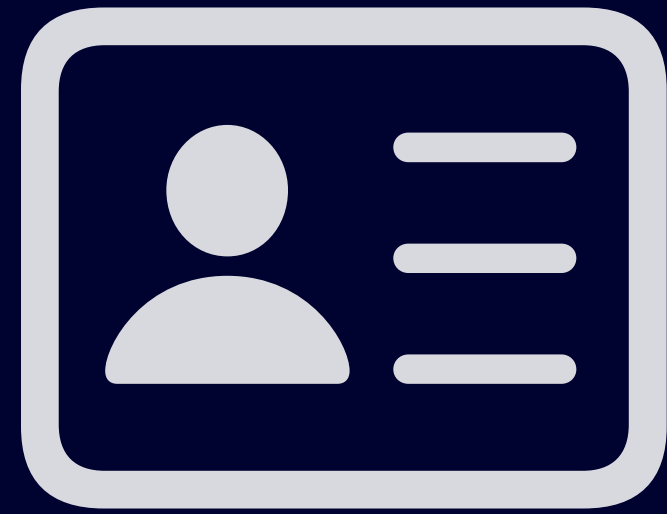
API keys

# Identification & Authentication

**API keys come in two flavors**

# Identification & Authentication

API keys come in two flavors

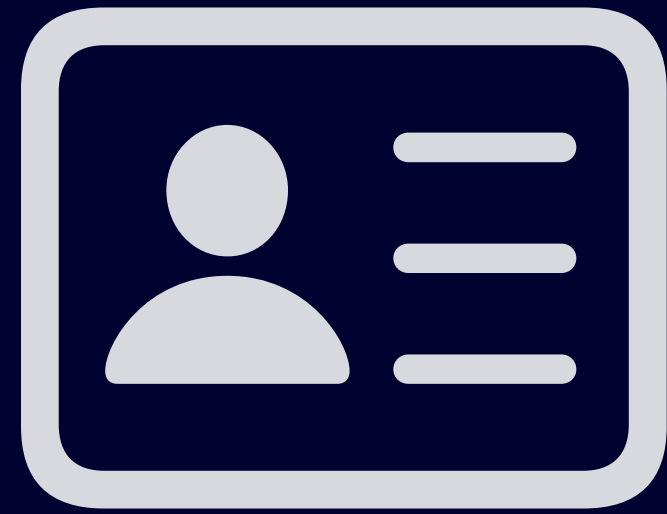


Identity key

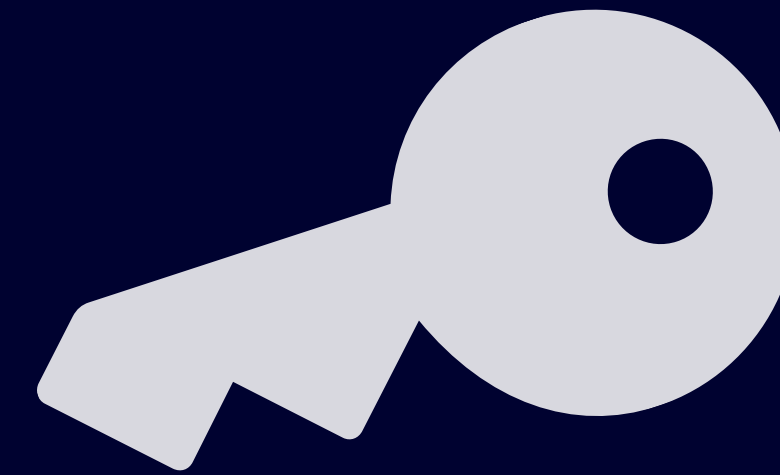


# Identification & Authentication

API keys come in two flavors



Identity key



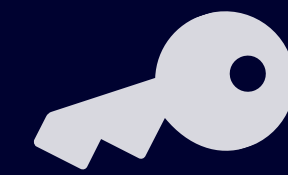
Authentication key

# Identification & Authentication

## API keys



Identity key



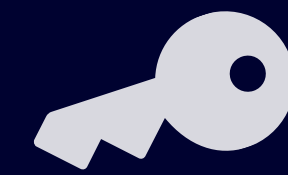
Authentication key

# Identification & Authentication

## API keys



Identity key



Authentication key

Identify...

your app

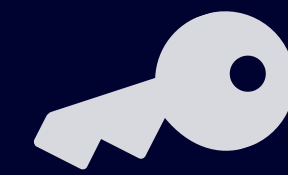
your account!

# Identification & Authentication

## API keys



Identity key



Authentication key

Identify...

your app

your account!

Data access

Restricted

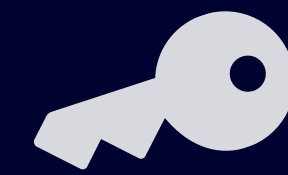
Unlimited

# Identification & Authentication

## API keys



Identity key



Authentication key

Identify...

your app

your account!

Data access

Restricted

Unlimited

Data update

Limited

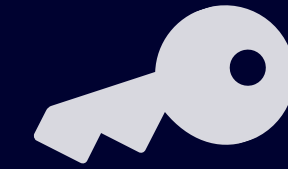
Unlimited

# Identification & Authentication

## API keys



Identity key



Authentication key

Identify...

your app

your account!

Data access

Restricted

Unlimited

Data update

Limited

Unlimited

Abuse can make...

limited to no damage

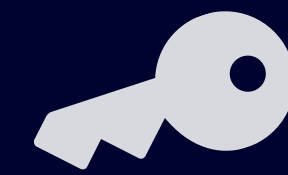
cost more than money

# Identification & Authentication

## API keys



Identity key



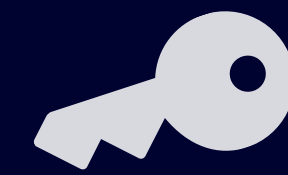
Authentication key

# Identification & Authentication

## API keys



Identity key



Authentication key

Commit them



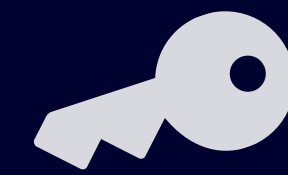


# Identification & Authentication

## API keys



Identity key



Authentication key

Commit them



Can afford to leak

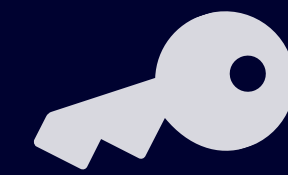


# Identification & Authentication

## API keys



Identity key



Authentication key

Commit them



Can afford to leak



Ship them

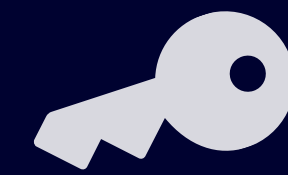


# Identification & Authentication

## API keys



Identity key



Authentication key

Commit them



Can afford to leak



Ship them



Pull them to your app

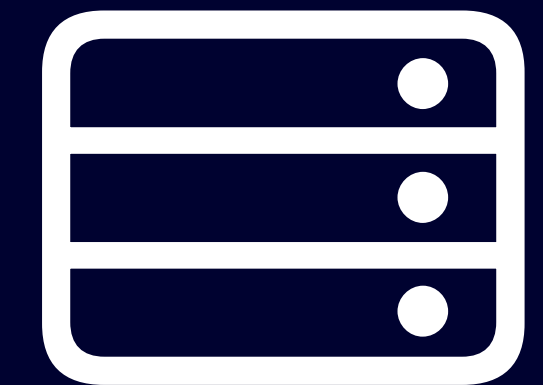


# Identification & Authentication

What to do with « Authentication » API keys?



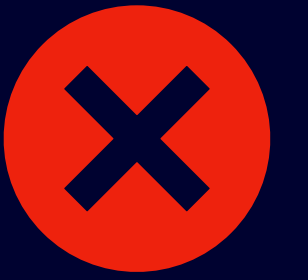
Your app



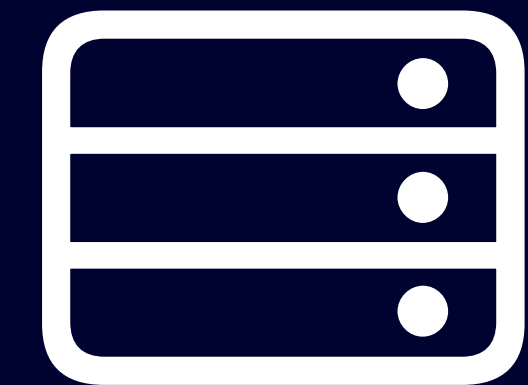
OpenAI

# Identification & Authentication

What to do with « Authentication » API keys?



Your app



OpenAI

# Identification & Authentication

What to do with « Authentication » API keys?



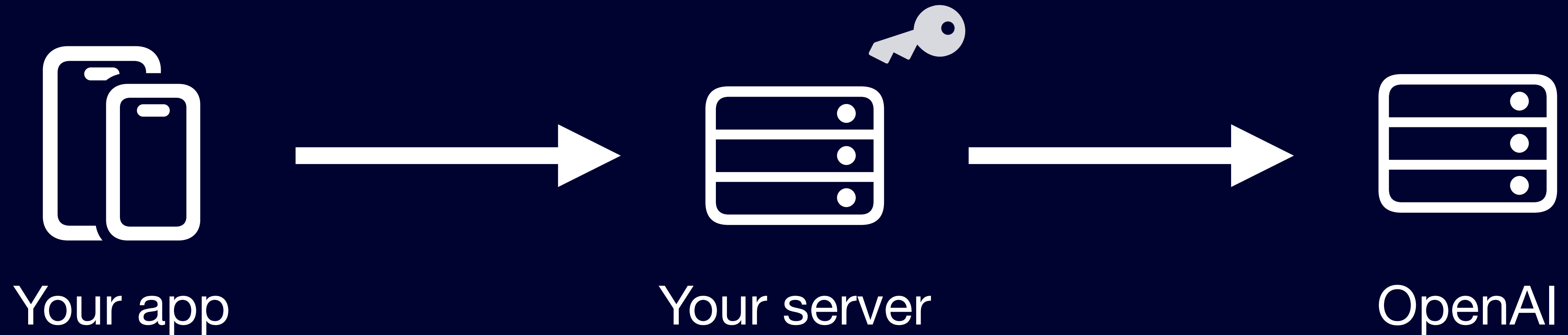
# Identification & Authentication

## What to do with « Authentication » API keys?



# Identification & Authentication

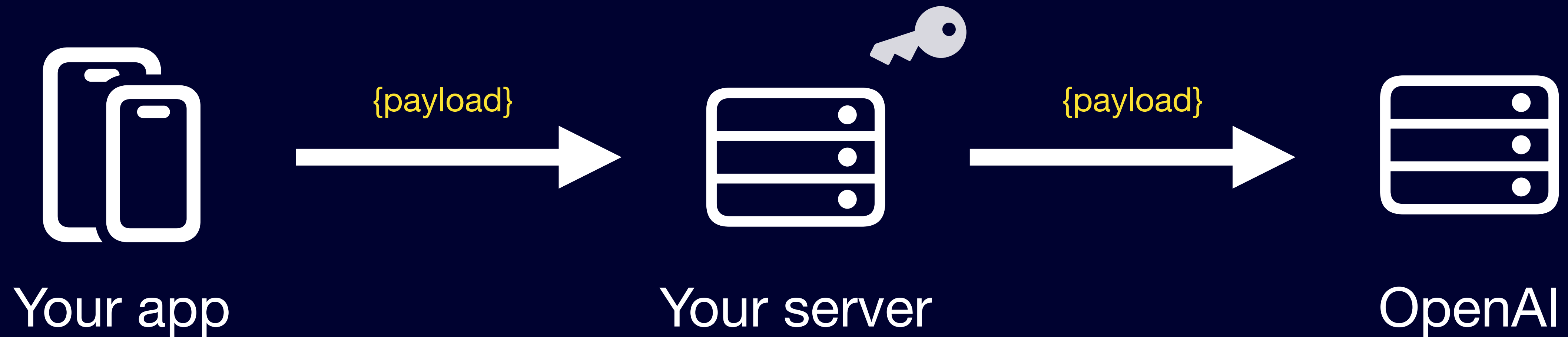
What to do with « Authentication » API keys?





# Identification & Authentication

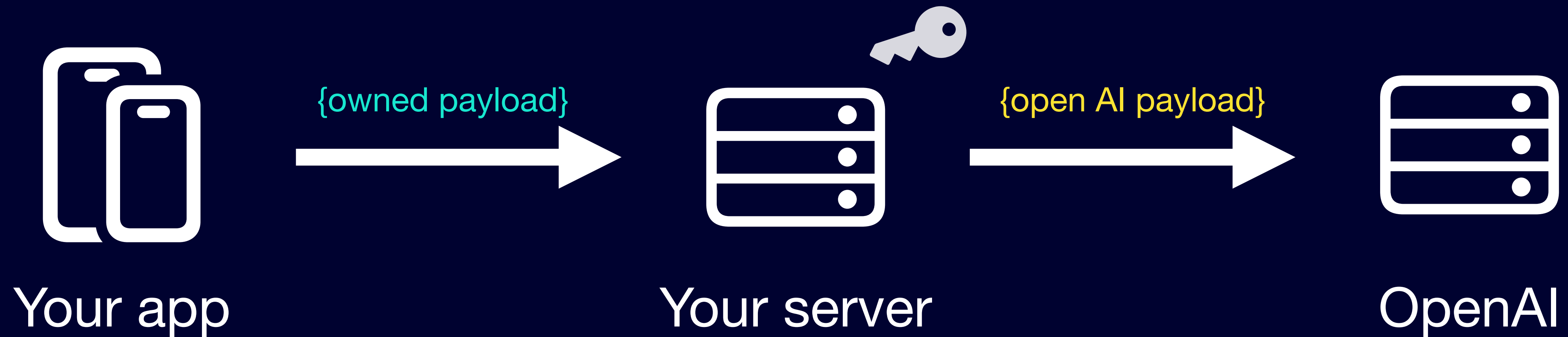
What to do with « Authentication » API keys?



**Don't proxy the payload, adding the key on the fly!**

# Identification & Authentication

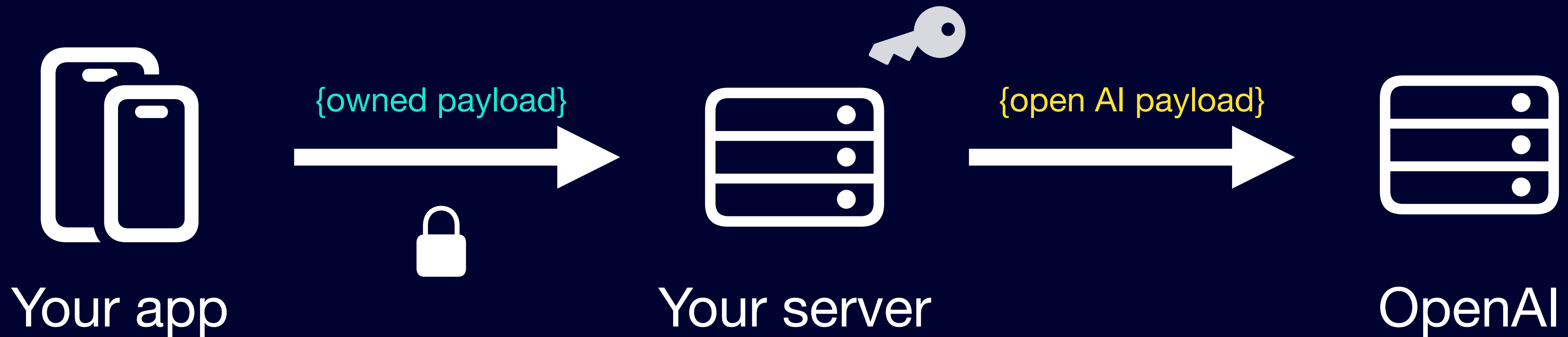
What to do with « Authentication » API keys?



Make your API reflect **your business logic**

# Identification & Authentication

What to do with « Authentication » API keys?



Authenticate your app if you can

# Identification & Authentication

Authenticate your app

# Identification & Authentication

## Authenticate your app

- Check the app authenticity 



# Identification & Authentication

## Authenticate your app

- Check the app authenticity 
  - DeviceCheck or AppAttest (iOS), Play integrity (Android)



# Identification & Authentication

## Authenticate your app

- Check the app authenticity 
  - DeviceCheck or AppAttest (iOS), Play integrity (Android)
- Authenticate your users 

# Identification & Authentication

## Authenticate your app

- Check the app authenticity 
  - DeviceCheck or AppAttest (iOS), Play integrity (Android)
- Authenticate your users 
  - Check in-app purchases server-side when applicable



# Identification & Authentication

About user authentication

# Identification & Authentication

## About user authentication

- User password is a **secret**

# Identification & Authentication

## About user authentication

- User password is a **secret**
  - If you have to store it → Keychain / KeyStore

# Identification & Authentication

## About user authentication

- User password is a **secret**
  - If you have to store it → Keychain / KeyStore
- Authentication is **hard to get right**

# Identification & Authentication

## About user authentication

- User password is a **secret**
  - If you have to store it → Keychain / KeyStore
- Authentication is **hard to get right**
  - Flow and security shall be enforced by your backend

# Identification & Authentication

## About user authentication

- User password is a **secret**
  - If you have to store it → Keychain / KeyStore
- Authentication is **hard to get right**
  - Flow and security shall be enforced by your backend
  - Identity providers are an opportunity → Sign in with Google / Apple

# Identification & Authentication

## About user authentication

- User password is a **secret**
  - If you have to store it → Keychain / KeyStore
- Authentication is **hard to get right**
  - Flow and security shall be enforced by your backend
  - Identity providers are an opportunity → Sign in with Google / Apple
  - ASWebAuthenticationSession / OpenID AppAuth-Android instead of your own webviews

# Identification & Authentication

## Automatic login or persistent authentication?



# Identification & Authentication

## Automatic login or persistent authentication?

- Bearer token, or session cookies are a **secret**

# Identification & Authentication

## Automatic login or persistent authentication?

- Bearer token, or session cookies are a **secret**
  - Session cookie is managed by the system

# Identification & Authentication

## Automatic login or persistent authentication?

- Bearer token, or session cookies are a **secret**
  - Session cookie is managed by the system
  - If you have to store them → Keychain / KeyStore

# Identification & Authentication

## Automatic login or persistent authentication?

- Bearer token, or session cookies are a **secret**
  - Session cookie is managed by the system
  - If you have to store them → Keychain / KeyStore
- Beware of **universal links!**

# Identification & Authentication

## Automatic login or persistent authentication?

- Bearer token, or session cookies are a **secret**
  - Session cookie is managed by the system
  - If you have to store them → Keychain / KeyStore
- Beware of **universal links!**
  - **Ask for user confirmation** for any action performed by a link opening your app

# Interlude #2



# Interlude #2

<http://fr.dev.alnpet.com>

<http://us1.dev.alnpet.com>



## Interlude #2



<http://fr.dev.alnpet.com>  
<http://us1.dev.alnpet.com>



## Interlude #2

No session cookie

<http://fr.dev.alnpet.com>  
<http://us1.dev.alnpet.com>



## Interlude #2

No session cookie

No bearer token

<http://fr.dev.alnpet.com>  
<http://us1.dev.alnpet.com>



## Interlude #2

No session cookie

No bearer token

No authentication mean at all...

<http://fr.dev.alnpet.com>  
<http://us1.dev.alnpet.com>



## Interlude #2



No session cookie

No bearer token

No authentication mean at all...

<http://fr.dev.alnpet.com>  
<http://us1.dev.alnpet.com>



## Interlude #2

No session cookie

No bearer token

No authentication mean at all...



**Instead of reverse-engineering the API**

Instead of reverse-engineering the API

**I reverse-engineered the machine**

<https://api.feedmypet.app/>

<https://github.com/Dean151/Aln-Symfony>

Instead of reverse-engineering the API

I reverse-engineered the machine

<https://api.feedmypet.app/>

<https://github.com/Dean151/Aln-Symfony>

And now, the original API is gone  
But my machine still work 😎



All these efforts to ensure  
the security of data transfer

All these efforts to ensure  
the security of data transfer

**What next?**

**A word on customer data**

**Storing sensitive data on device?**

Storing sensitive data on device?

Encrypt it!

# Storing sensitive data on device?

**Encrypt it!**

Symmetric key

—

Generated when needed

—

Stored in Keychain / KeyStore

**Storing sensitive data on server?**

Storing sensitive data on server?

Encrypt it!



# Storing sensitive data on server?

## Encrypt it!

Symmetric key

—

Generated when needed

—

Stored in a KMS

# Don't require data access on the server?

This one will surprise you

**Don't require data access on the server?**

**This one will surprise you**

**End-to-end encryption**

# Don't require data access on the server?

This one will surprise you

## End-to-end encryption

Customer data can't leak if you don't own the keys...

**“It’s useless to secure the door if you leave the window open”**

# Thank you!

Slides:

<https://thomasdurand.fr/mim-2024>

<https://thomasdurand.fr>

[@deanatoire@mastodon.social](https://mstdn.social/@deanatoire)

[@deanatoire@threads.net](https://www.threads.net/@deanatoire)

[@deanatoire@twitter.com](https://twitter.com/deanatoire)