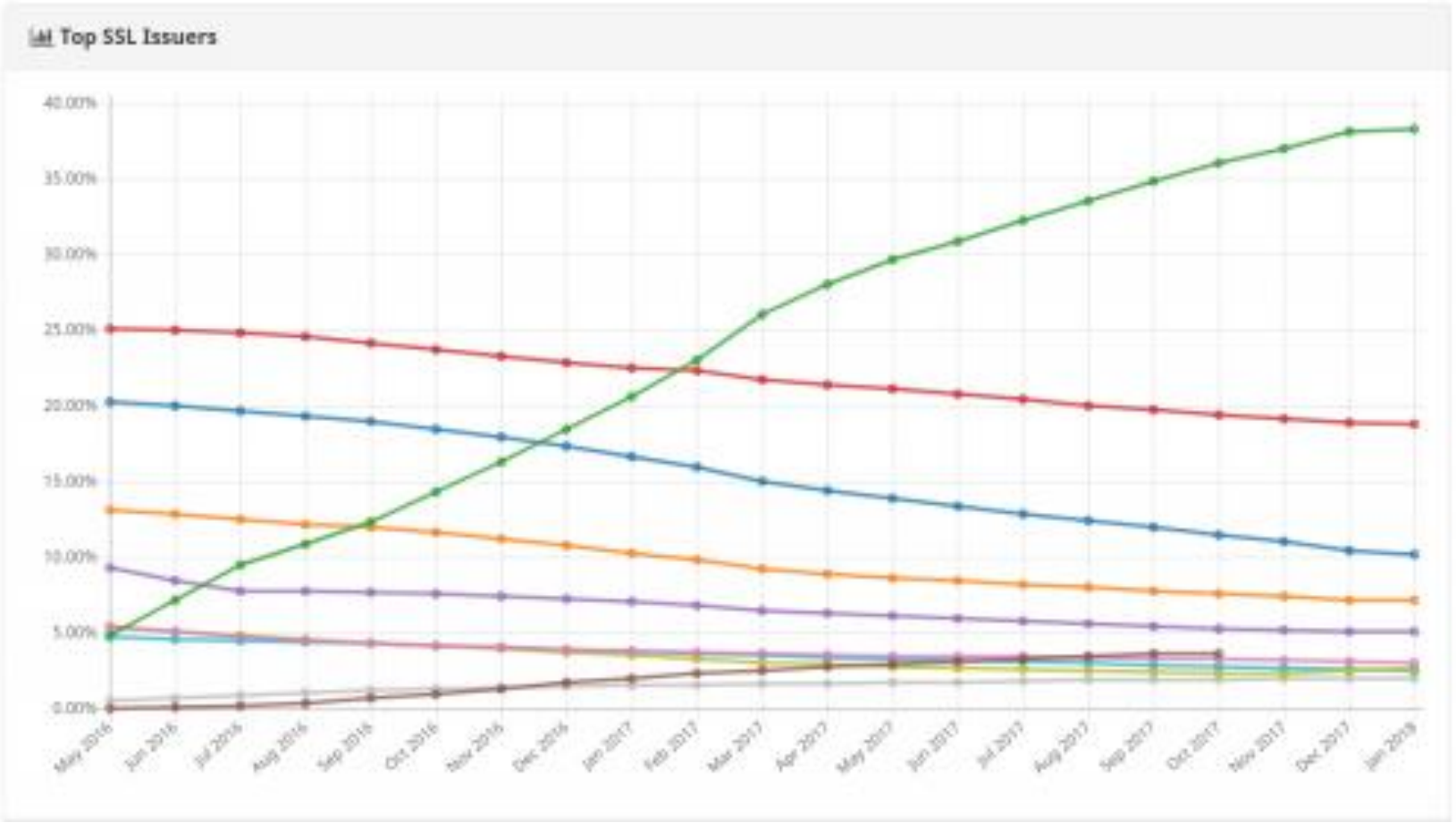
LE PROTOCOLE ACME

ORIGINES (1)

En 2015 est lancé Let's Encrypt

- But : généraliser l'usage de connexions sécurisées sur Internet
  - Couteuses (peu ou pas d'AC gratuites – StartCom de 2009 à 2016, CloudFlare – Universal SSL - en 2014)
- Service proposé
  - Fourniture de certificats de classe 1 (DV) gratuits et valides 3 mois
  - Via un protocole : ACME
  - Et son client Open Source: certbot (ex letsencrypt.sh)
- Fondé par **Internet Security Research Group** (**ISRG**)
  - Organisme à but non lucratif
  - Participants: Cisco, University of Michigan, Mozilla, ACLU, CoreOS, Facebook, and the Electronic Frontier Foundation

ORIGINES (2): LET'S ENCRYPT



#	SSL Issuer	Percentage
1	Let's Encrypt	38.31%
2	COMODO CA Limited	18.8%
3	GeoTrust Inc.	10.23%
4	GoDaddy.com	7.19%
5	GlobalSign nv-sa	5.09%
6	cPanel	3.68%
7	Symantec Corporation	3.06%
8	DigiCert Inc	2.71%
9	thawte	2.54%
10	Amazon	1.99%

WHAT IS THAT ?

Automatic Certificate Management Environment

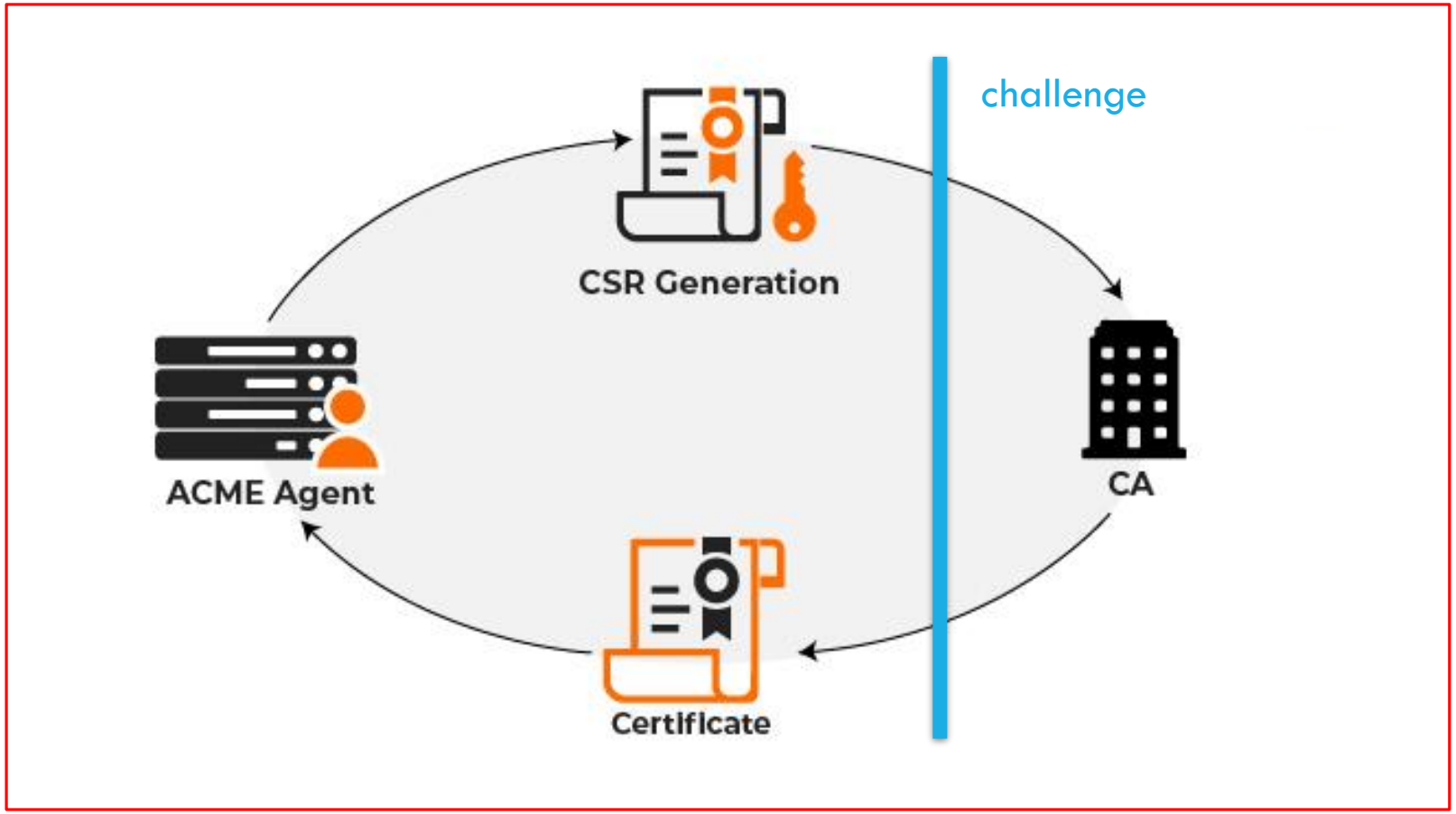
Protocole de communication avec des AC (CA)

Permettant de générer des certificats à la demande

ACME v2 est aujourd'hui un standard: RFC8555

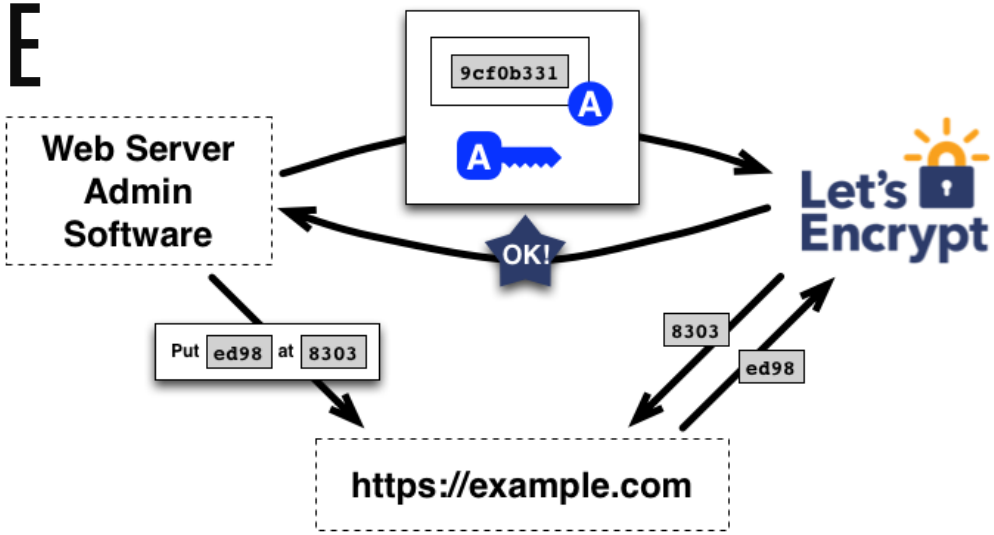
FONCTIONNEMENT DU PROTOCOLE: CHALLENGE

standardisé



Échanges en JSON

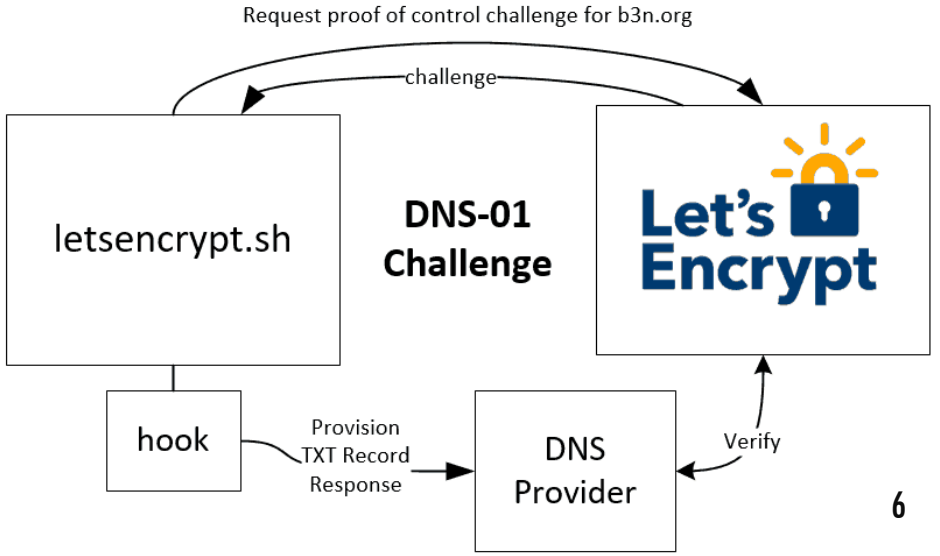
DIFFÉRENTS TYPES DE CHALLENGE



Dans le cas de Let's Encrypt

- HTTP-01: le CA valide un fichier hébergé sur le domaine (en http)
  - http://<DOMAIN>/.well-known/acme-challenge/<TOKEN>

- DNS-01: le CA valide un enregistrement DNS correspondant à une clé locale
  - _acme-challenge.<DOMAIN> IN TXT "<TOKEN>"
  - Permet de générer des wildcard (http-01 ne le permet pas)



ET CONCRÈTEMENT (1) ?

On installe le client certbot

On demande le certificat au provider

On résout le challenge

On obtient le certificat

Exemple avec Let's Encrypt et certbot

… sur un sous-domaine de capitoul (merci ☺)

[DEMO OFFLINE]

apt install certbot

Version standalone (certbot lance serveur web autonome)

certbot certonly --standalone -d testacme1.capitoul.org --email une-adresse-mail@un-domaine.fr --agree-tos

Version webroot (utilise un serveur existant / racine web)

apt install apache2
service apache2 start
certbot certonly --webroot -w /var/www/html/ -d testacme2.capitoul.org --email une-adresse-mail@un-domaine.fr --agree-tos

=> Les certificats sont alors disponibles dans /etc/letsencrypt/live/[domaine]

ET CONCRÈTEMENT (2) ?

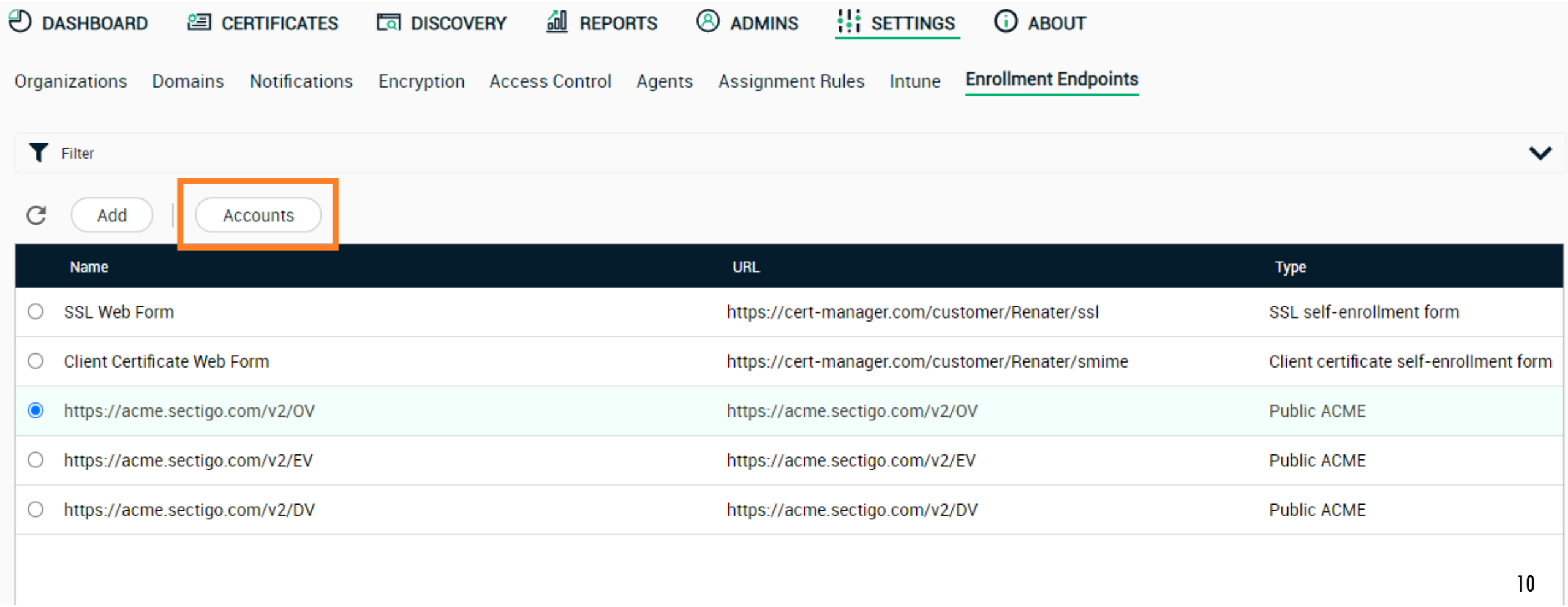
Certbot: challenge HTTP-01

- standalone (certbot lance serveur web autonome)
- webroot (utilise un serveur existant / racine web)
- nginx (va bidouiller les fichiers nginx)

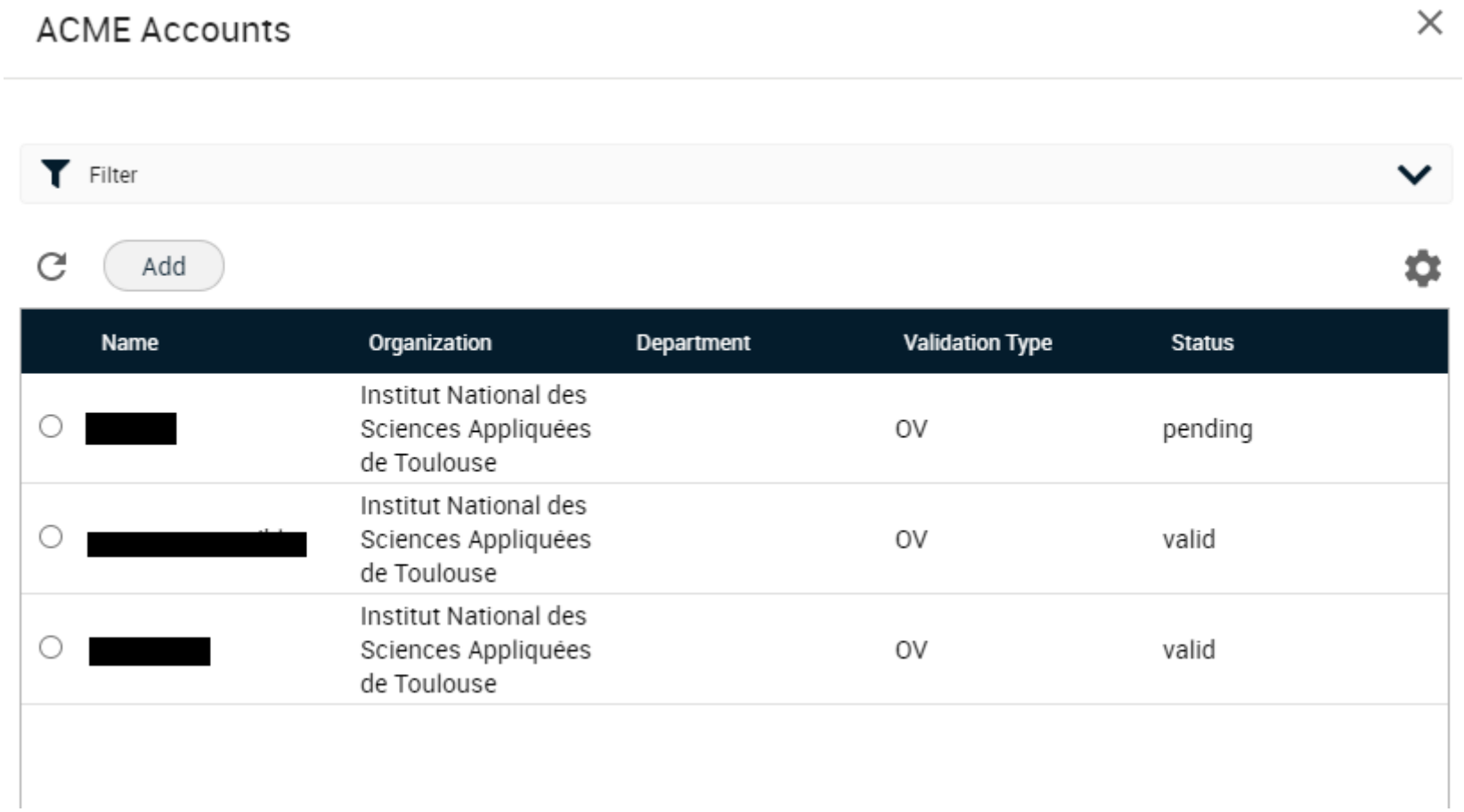
Et pour l'ESR ?

- Marché TCS / Géant: Sectigo
  - Le challenge est une authentification du client
    - Gestion du compte uniquement avec ACME v2
  - Certificat OV (EV non testé => double-validation ?)

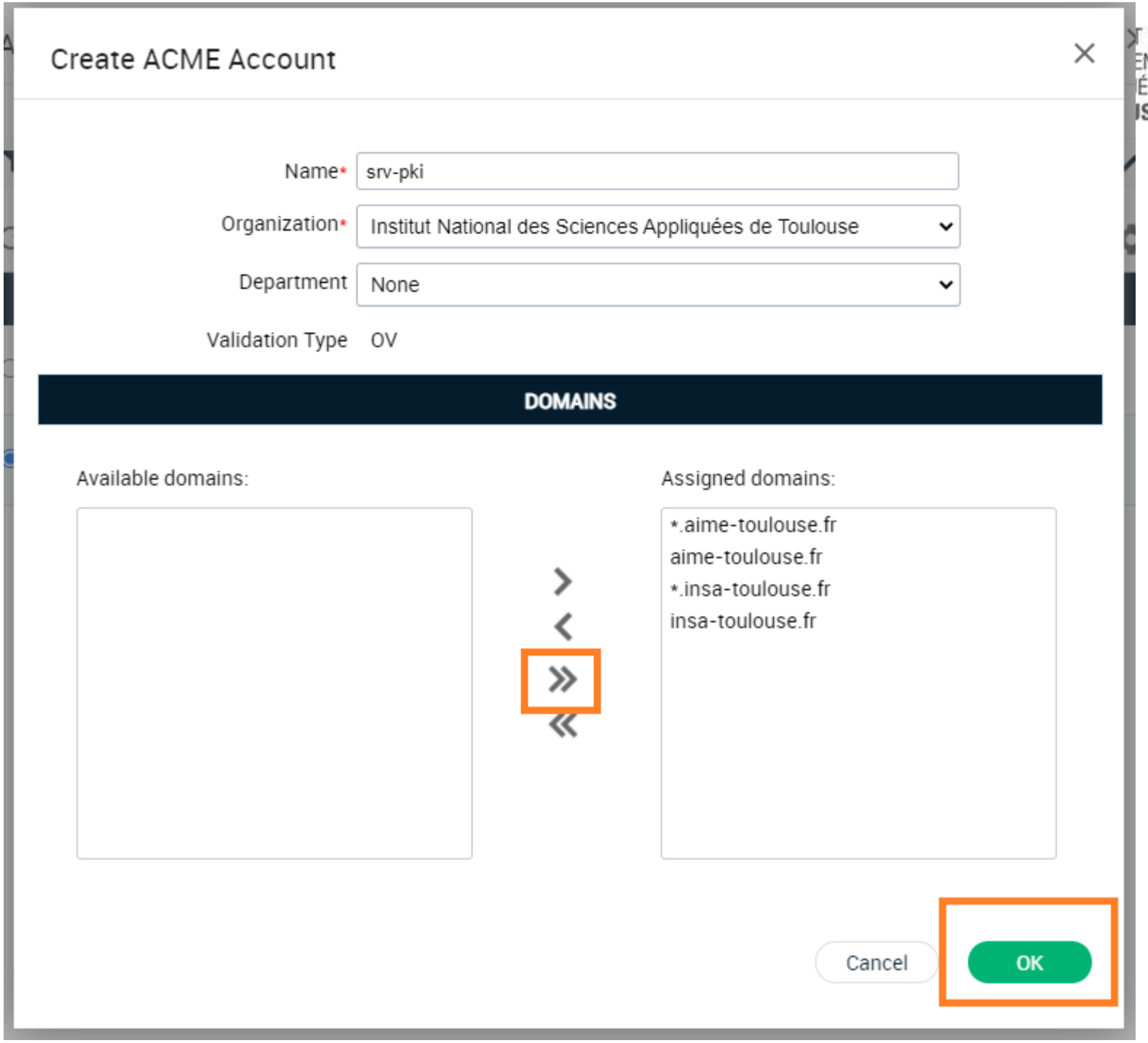
MISE EN PLACE AVEC SECTIGO (1)



MISE EN PLACE AVEC SECTIGO (2)



SECTIGO (3)


Create ACME Account
Name*
srv-pki
Organization*
Institut National des Sciences Appliquées de Toulouse
Department
None
Validation Type
OV
DOMAINS
Available domains:
Assigned domains:
*.aime-toulouse.fr
aime-toulouse.fr
*.insa-toulouse.fr
insa-toulouse.fr
Cancel
OK

MISE EN PLACE AVEC SECTIGO (4)

ACME Account details: srv-pki

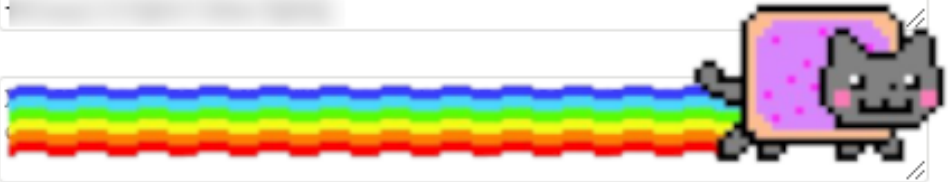
EXTERNAL ACCOUNT BINDING

ACME URL: https://acme.sectigo.com/v2/OV

Account ID: [redacted]

Key ID: [redacted]

HMAC Key: [redacted]



Instructions on how to use account:

When ACME client (certbot) is initialized, we specify ACME URL (--server), MAC Key (--eab-hmac-key), Mac ID (--eab-kid).
Example command line: sudo certbot certonly --standalone --

Close

INTÉGRATION DANS LE SI

Différences par rapport à Let's Encrypt

- Pas de réel challenge, accès libre à la génération
- Les certificats sont générables partout où est stockée la clé
- Attention à la protection de cette clé: gestion de PKI à mettre en place

- Client certbot:
  - À l'INSA: script qui encapsule et déploie les certificats dans tous les formats (via SCP)
    - Depuis un serveur de déploiement (a accès à tous les serveurs)
    - Les certificats sont sauvegardés en local

```
CRT: /etc/ssl/www/www.crt
KEY: /etc/ssl/www/www.key
CHAIN: /etc/ssl/www/www.chain
CRT+CHAIN: /etc/ssl/www/www.chained.crt
JKS: /etc/ssl/www/www.jks (mdp: changeit)
--
Configuration Apache

          SSLCertificateFile    /etc/ssl/www/www.crt
          SSLCertificateKeyFile /etc/ssl/www/www.key
          SSLCACertificateFile  /etc/ssl/www/www.chain

--
Configuration Nginx

          ssl_certificate     /etc/ssl/www/www.chained.crt;
          ssl_certificate_key /etc/ssl/www/www.key;

--
Configuration HAProxy

          bind xxx:443 ssl crt /etc/ssl/www/www.full.pem
```

```
# cert_from_acme
Usage: /usr/local/bin/cert_from_acme shortname [shortaliasnames...]
# cert_from_acme www messagerie moncompte
```

INTÉGRATION DANS LE SI: CERTBOT

```
for san in `echo $* | sed -e 's/,/ /'`; do
    args="$args -d ${san}.$suffix"
done

certbot-auto certonly \
        --standalone \
        --non-interactive \
        --agree-tos \
        --email [redacted] \
        --server https://acme.sectigo.com/v2/OV \
        --eab-kid "$(cat /usr/local/etc/sectigo/key)" \
        --eab-hmac-key "$(cat /usr/local/etc/sectigo/hmac)" \
        $args \
        --cert-name $commonName
```

Les deux clés permettent d'initialiser le compte qui sera stocké dans /etc/letsencrypt/accounts/acme.sectigo.com/v2/OV

Ces deux clés ne sont plus nécessaires ensuite

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

INTÉGRATION DANS LE SI: ANSIBLE ACME (1)

```
- name: Generate an OpenSSL private key with the default values (4096 bits, RSA)
  community.crypto.openssl_privatekey:
    path: /etc/ssl/private/acme-account.pem
  connection: local

- name: Change account's key to the one stored in the variable new_account_key
  community.crypto.acme_account:
    acme_version: 2
    account_key_src: "{{ acme_key }}"
    state: present
    acme_directory: "{{ acme_directory }}"
    external_account_binding:
      key: "{{ acme_hmac_key }}"
      kid: "{{ acme_key_ID }}"
      alg: HS256
  connection: local
```

INTÉGRATION DANS LE SI: ANSIBLE ACME (2)

```
- name: Create a directory if it does not exist
  ansible.builtin.file:
    path: "{{ certs_path }}"
    state: directory
    mode: '0700'

- name: Generate an OpenSSL private key with the default values (4096 bits, RSA)
  community.crypto.openssl_privatekey:
    path: "{{ certs_path }}{{ crt_common_name }}.pem"
    type: ECC
    curve: secp256r1
  connection: local

- name: generate csr
  openssl_csr:
    path: "{{ certs_path }}/{{ crt_common_name }}.csr"
    privatekey_path: "{{ certs_path }}{{ crt_common_name }}.pem"
    common_name: "{{ crt_common_name }}"
    subject_alt_name: "DNS:{{ crt_subject_alt_name | join(',DNS:') }}"
```

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

INTÉGRATION DANS LE SI: ANSIBLE ACME (3)

```
- name: create acme challenge
  community.crypto.acme_certificate:
    acme_version: 2
    account_key_content: "{{ acme_key_contents }}"
    csr: "{{ certs_path }}/{{ crt_common_name }}.csr"
    dest: "{{ certs_path }}/{{ crt_common_name }}.crt"
    acme_directory: "{{ acme_directory }}"
  register: challenge


- name: create acme challenge
  community.crypto.acme_certificate:
    acme_version: 2
    account_key_content: "{{ acme_key_contents }}"
    csr: "{{ certs_path }}/{{ crt_common_name }}.csr"
    dest: "{{ certs_path }}/{{ crt_common_name }}.crt"
    fullchain_dest: "{{ certs_path }}{{ crt_common_name }}.full.crt"
    acme_directory: "{{ acme_directory }}"
    chain_dest: true
    data: "{{challenge}}"
```

INTÉGRATION DANS LE SI

Ansible vs certbot

- Pour la génération et le cycle de vie: ++certbot
- Pour le déploiement: les deux conviennent
  - Ansible permettra de mettre en place la copie des certificats (éventuellement générés via certbot), la supervision
  - Sur quelle machine lancer l'opération: question du stockage de la clé privée
  - Rajout d'une variable dans host_vars:

```
-
certificates:
  www:
    - aliases: [messagerie, moncompte]
```

.. vs Sectigo REST API ? (exemples de playbook sur le net)

Cycle de vie

- certbot renew (cron) + envoi des certificats
  - non encore mis en place chez nous

LE VRAC

On peut aussi révoquer (certbot revoke)

Let's Encrypt n'est pas la seule AC gratuite: Buypass Go SSL (Norvègienne), ZeroSSL (Australienne)

ACME v1 et v2 sont incompatibles mais cohabitent (la gestion de compte est uniquement dans le v2)

- Let's Encrypt a désactivé ACME v1 pour les nouveau domaines

Il existe d'autres clients ACME:

- Clients
  - certbot (Python)
  - getSSL (bash)
  - Posh-ACME (PowerShell)
  - ACMESharp (.NET)
- Intégrations / Librairies
  - Module Apache mod_md
  - Nginx
  - Java PAJC
  - Ansible
- https://letsencrypt.org/fr/docs/client-options/

=> non testés, ne se prêtent pas forcément à notre cas d'usage (pas de DNS/HTTP challenge)

Établissements intéressés par un rôle ansible Sectigo ?

FIN

~~systemctl stop capitoul-acme.service~~

systemctl kill capitoul-acme.service

systemctl start capitoul-qaa.service