

CERT-BROKER

Certificats ACME via DNS01, distribués par Git

Pierre Gambarotto



INSTITUT DE MATHÉMATIQUES
de TOULOUSE



CERT-BROKER

- obtenir des certificats par le challenge DNS01 d'acme
- les rendre disponibles dans un dépôt git par ssh

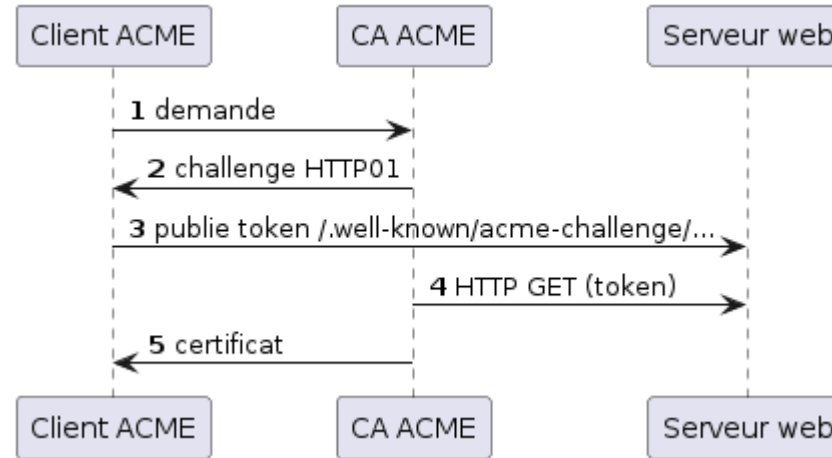
Utilité :

- pour les certificats * et les machines non exposées
- Livraison simple aux utilisateurs et à la CI

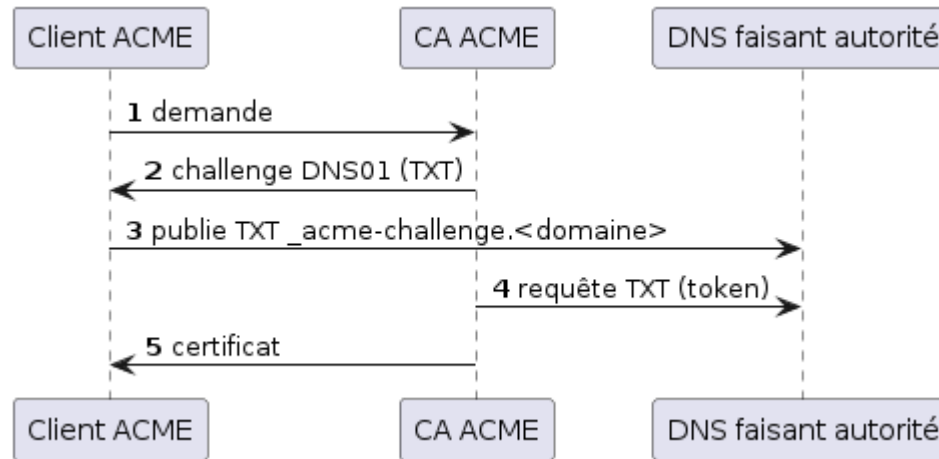
ACME POUR LES CERTIFICATS

- obtenir des certificats automatiquement via un challenge
- les renouveler de manière similaire
- popularisé par [Let's Encrypt](#) de l'[Internet Security Research Group](#)

HTTP01 : PREUVE PAR UNE URL HTTP SUR LE SERVEUR



DNS01 : PREUVE PAR UN ENREGISTREMENT DNS TEMPORAIRE



DNS01 : QUAND ET POURQUOI

- Wildcard : seul DNS01 permet *.domaine
- Machines non exposées (filtrage, IP privée, pas de HTTP public)

MODIFIER LE DNS POUR PUBLIER LA RÉPONSE AU CHALLENGE

- à la main
- par API DNS : token api sur des serveurs 🤖
- agnos !

AGNOS

<https://github.com/krtab/agnos>

- Client ACME + serveur DNS minimal
- Publie la réponse DNS01 automatiquement
- Ne répond qu'aux requêtes de vérification TXT
_acme-challenge.<...>

L'idée phare : le DNS de la zone va déléguer la résolution pour les requêtes de vérification au serveur agnos.

AGNOS EN TANT QUE SERVEUR DNS

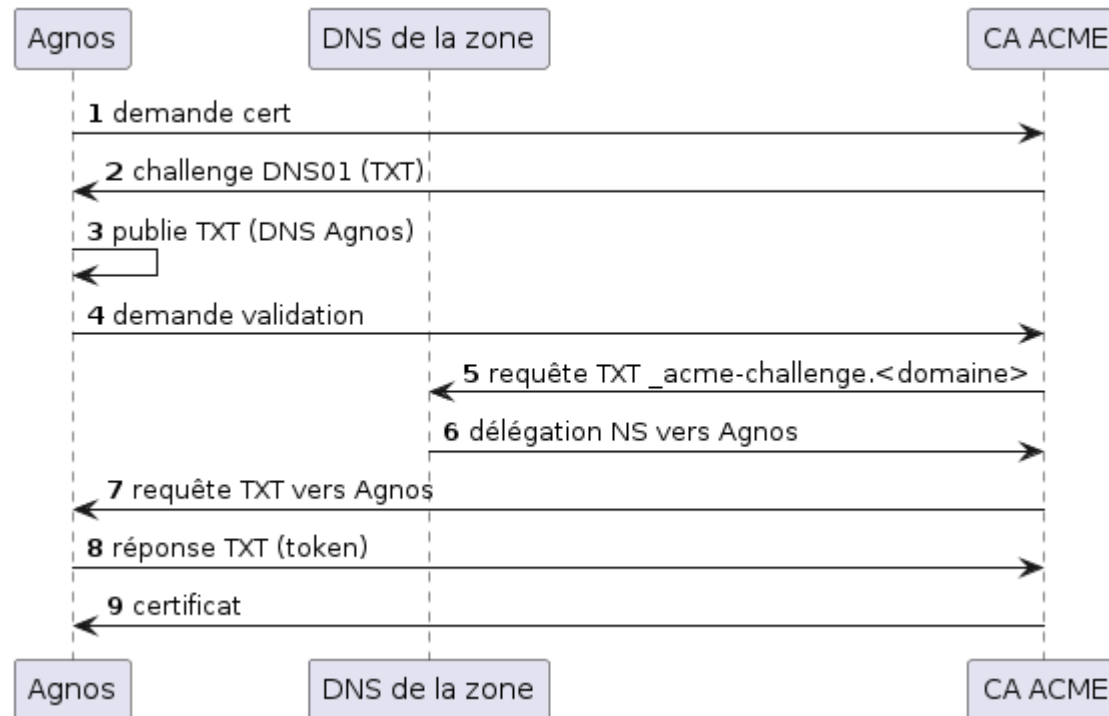
Exemple pour la zone example.org : déléguer
_acme-challenge vers agnos.other.fr

```
; zone example.org
; cert demandé : *.example.org
_acme-challenge.example.org.  IN NS  agnos.other.fr.

; cert demandé : www.example.org
_acme-challenge.www.example.org.  IN NS  agnos.other.fr.

; zone other.fr
agnos.other.fr.                IN A    203.0.113.10
```

AGNOS : SCHÉMA



BILAN AGNOS

Pour gérer un certificat, l'utilisateur doit :

- publier dans le DNS de la zone une délégation fixe
- Agnos fait le reste (challenge DNS01 et renouvellements)
- Les certificats se retrouvent sur le serveur Agnos

Pour gérer la distribution :

- Couplage à un serveur git pour un accès simple par git clone -> Forgejo

FORGEJO

<https://forgejo.org/>

fork de gitea, un seul petit binaire à déployer

configuration verrouillée : seul le serveur peut créer
des repo

partage des repo par clefs ssh uniquement

CERT- BROKER.MATH.CNRS.FR

agnos + forgejo piloté par CI DNS Plm

DNS PLM

- par zone dns gérée, un dépôt sur plmlab, le gitlab de la PLM
 - mettre à jour : créer une Merge Request
- > un pipeline avant le merge : vérifier la validité
- > un pipeline après le merge : appliquer les changements sur le serveur
- `dns0.math.cnrs.fr`

EXAMPLE

Dépôt `math.univ-toulouse.fr`

- `math.univ-toulouse.fr.data`: zone au format bind9
- `metadata.json`

```
{
  "zones": [
    "math.univ-toulouse.fr"
  ],
  "contacts": [
    "Pierre Gambarotto <pierre.gambarotto@math.univ-toulouse.fr>",
    "Ludovic Fabre <ludovic.fabre@math.univ-toulouse.fr>",
    "Mathys Astruc <mathys.astruc@math.univ-toulouse.fr>"
  ]
}
```

AVANT LE MERGE

```
check-zone-syntax:  
  tags: [merge-zone]      # runners partagés  
  stage: test  
  script:  
    - /bin/check-zone-syntax  
  only:  
    - merge_request
```

```
merge-new-zone-version:  
  tags: [merge-zone]      # runners partagés  
  stage: merge  
  only:  
    - merge_request        # auto accept MR  
  script:  
    - echo "apiToken is ${apiToken}"  
    - /bin/merge-new-zone-version
```



Jobs faits par des runners partagés, MR automatique

APRÈS LE MERGE

```
update-master:  
  tags:  
    - update-master # runner shell sur dns0.math.cnrs.fr  
  stage: dns0  
  only:  
    - main  
  script:  
    - /run/current-system/sw/bin/update-master
```

Job fait par un runner en mode shell sur le serveur
dns0.math.cnrs.fr

ADAPTATION POUR CERT-BROKER

Pour chaque certificat à générer :

- agnos : enregistrement NS : à mettre dans le fichier de zone
- git clone : clef publique ssh : à mettre dans metadata.json

EXEMPLE

générer un certificat pour `test.math.univ-toulouse.fr`

```
_acme-challenge.test      300  IN  NS  cert-broker.math.cnrs.fr.
```

```
{
  "cert-broker": {
    "mycert": {
      "email": "pierre.gambarotto@math.univ-toulouse.fr",
      "domains": ["test.math.univ-toulouse.fr"],
      "sshpubkeys": [
        "ssh-ed25519 AAAAC3NzaC1...vWNf/zJU"
      ]
    }
  }
}
```

-> `math.univ-toulouse.fr/mycert` projet
forgejo généré

CI

2 jobs sur le pipeline *après* le merge

- `cert-broker-check` : valide le format, runner partagé
- `cert-broker-deploy` : appelle agnos, génère le dépôt forgejo, runner shell sur `cert-broker.math.cnrs.fr`

variable `CERT_BROKER_SERVER` définie au niveau du groupe regroupant les zones

CERT-BROKER-CHECK

```
cert-broker-check:
  # image générée par nix
  image: "registry.xxx/dns/ci/cert-domain-check:latest"
  stage: test
  tags: [merge-zone] # runners partagés
  script:
    - cert-domain-check -o . . || true # on n'échoue jamais !!
  only:
    - main
  artifacts:
    reports:
      dotenv: cert-broker.env # CERT_BROKER_ENABLE = "true" si o
    paths:
      - cert-broker-errors.txt # erreurs si pas ok
  when: always
```

CERT-BROKER-DEPLOY

```
cert-broker-deploy:
  stage: cert-broker
  tags: [cert-broker] # runner shell sur cert-broker.math.cnrs.fr
  only:
    - main
  needs: [cert-broker-check]
  script:
    - |
      if [ "$CERT_BROKER_ENABLED" != "true" ]; then
        echo "cert-broker check failed or absent, skipping"
        exit 0
      fi
    cert-domain .
```

DÉMO

CONCLUSION

- gestion par CI : un job pour valider les données, un job pour appliquer
- forgejo : gitlab prend garde à toi
- en test !