向き合おう、DNSとサーバー証明書

～DNSとサーバー証明書の最近の関係を踏まえ、DNS運用者がすべきこと～ ランチのおともにDNS

2017年11月30日

Internet Week 2017 ランチセミナー

株式会社日本レジストリサービス（JPRS）

森下 泰宏・島田 直人

講師自己紹介

- 森下 泰宏（もりした やすひろ）
  - 所属：JPRS 技術広報担当
  - 主な業務内容：ドメイン名・DNSに関する技術広報活動全般
  - 一言：今年はRFC 1034・1035の発行から30周年です
- 島田 直人（しまだ なおと）
  - 所属：JPRS システム部
  - 主な業務内容：DNSSECの運用、オフィスシステムの運用
  - 一言：JPNICと同じ年に生まれました

本日の内容

1. DNSと証明書の最近の関係
2. DNSを用いた証明書関連技術
   - 2.1 CAAレコード
   - 2.2 自動証明書管理環境（ACME）における、DNS経由での認証
3. 最近の関係を踏まえ、DNS運用者がすべきこと

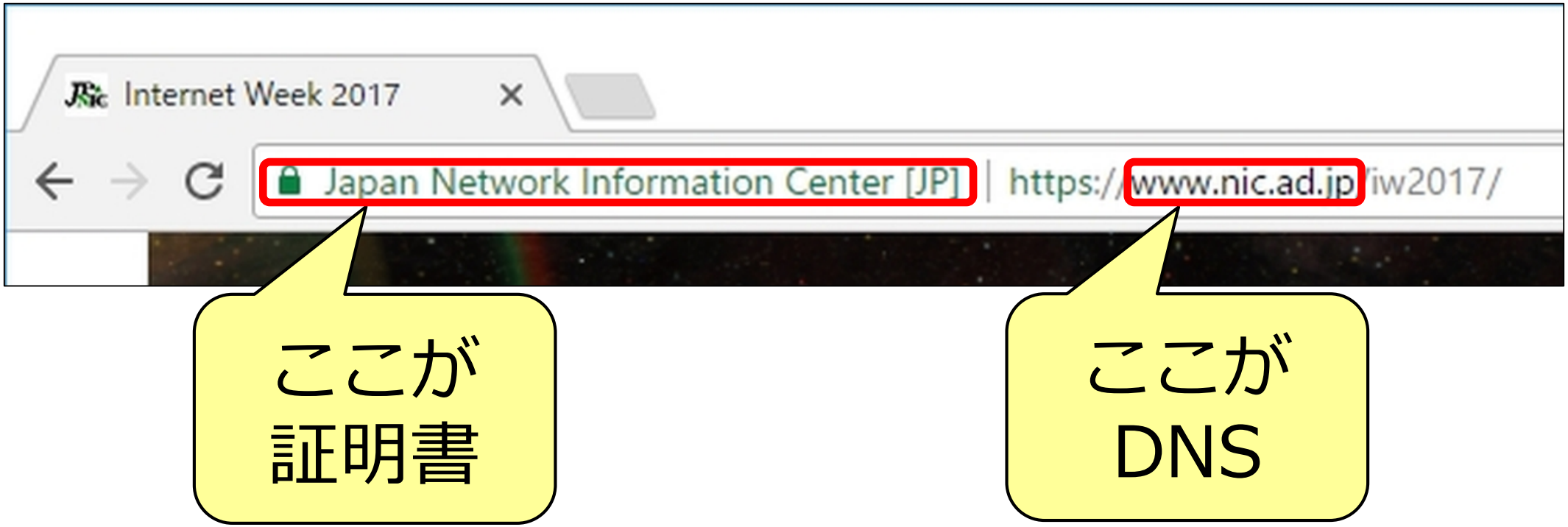
注：本資料では証明書を電子証明書、特にTLSの「サーバー証明書」の意味で使用します

本日は1.と3.を森下が、2.を島田が担当します

1. DNSと証明書の最近の関係

Internet Week 2017トップページ

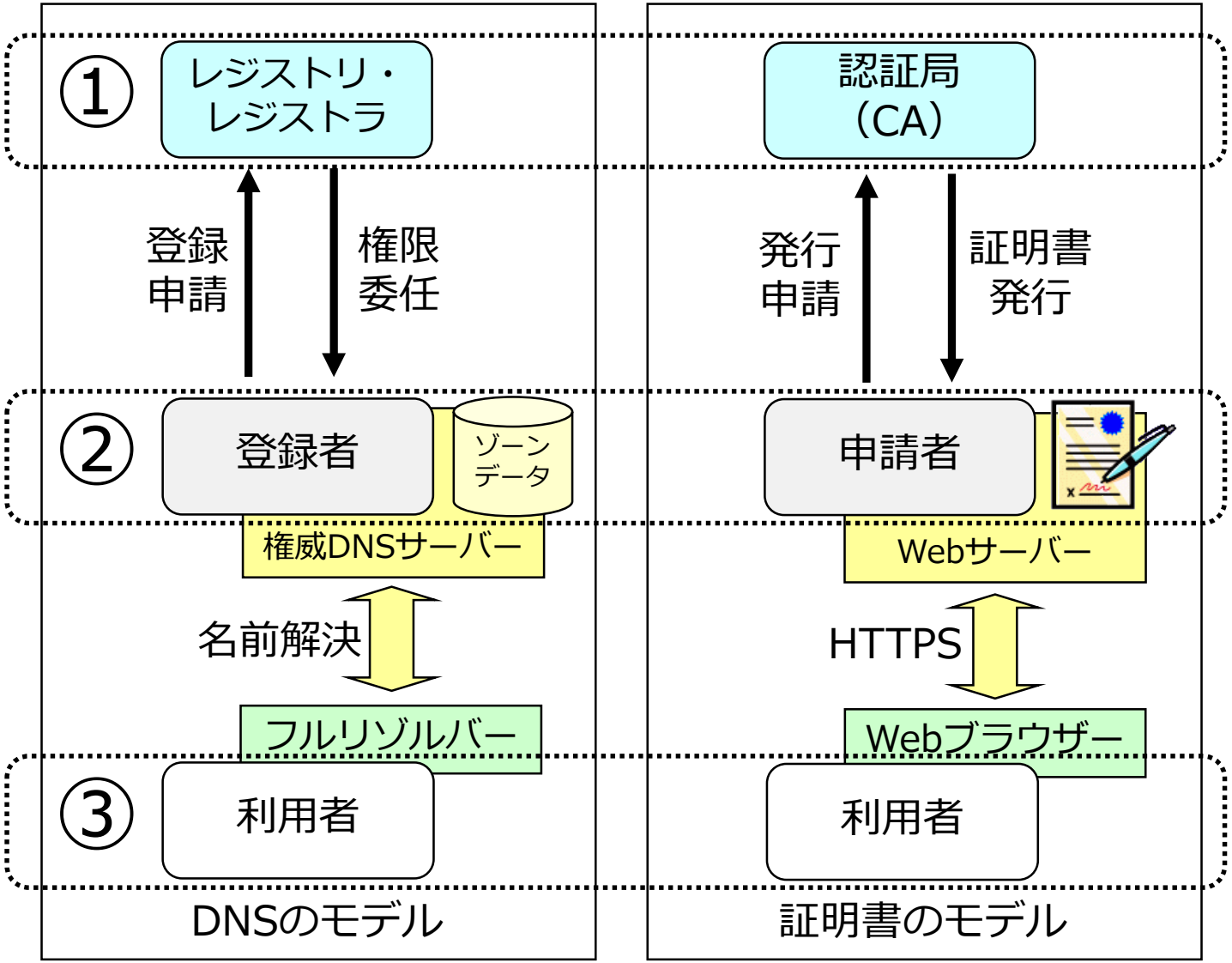
アドレスバーのDNSと証明書



- Webブラウザーのアドレスバーで共存している
- インターネットの根幹にかかわる、重要な役割を担っている

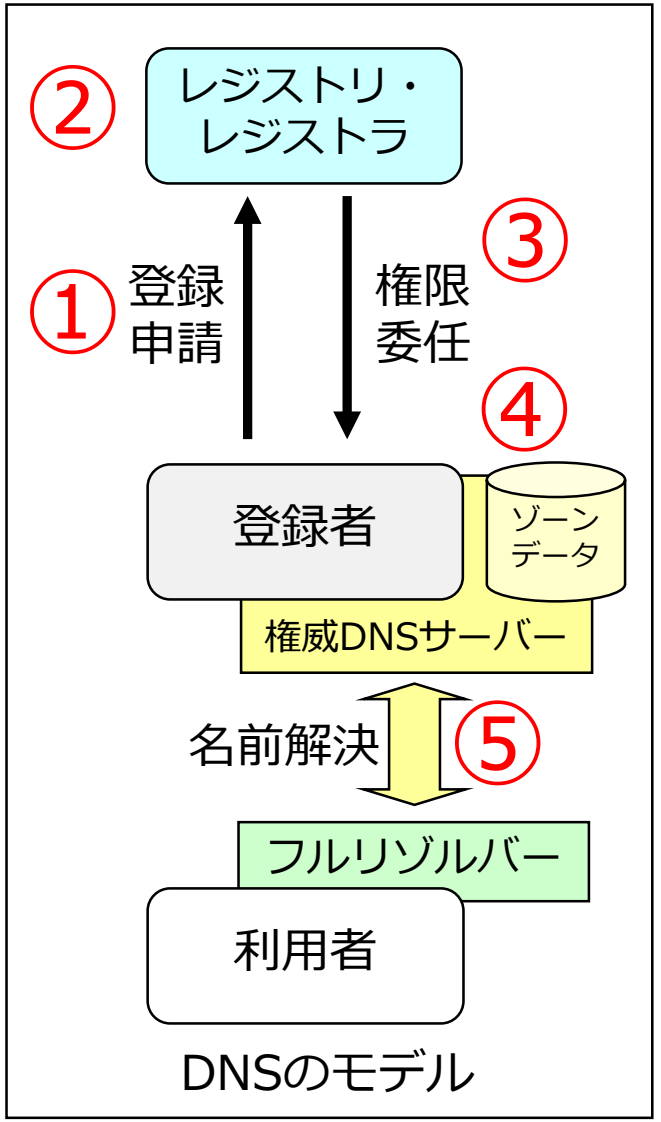
DNSと証明書―利用の形態

- 利用の形態、及び申請から利用までの流れに類似性がある
- 利用の形態
  - ① リソースを提供する人
  - ② リソースを申請・設定する人
  - ③ 設定されたリソースを利用する人

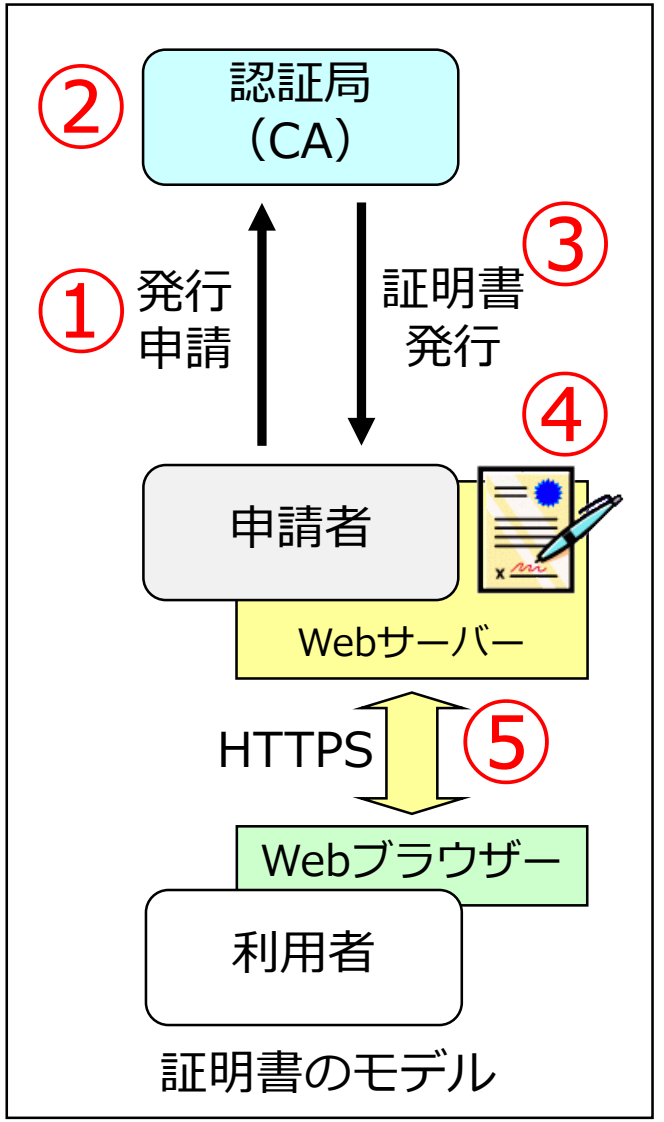


DNSと証明書―申請から利用までの流れ

- 申請から利用までの流れ
  - ① 申請
    - 必要なリソースを申請
  - ② 確認
    - 所定の方法で要件を確認
  - ③ 提供
    - 申請されたリソースを提供
  - ④ 設定
    - サーバーにリソースを設定
  - ⑤ 利用
    - 所定の方法で利用



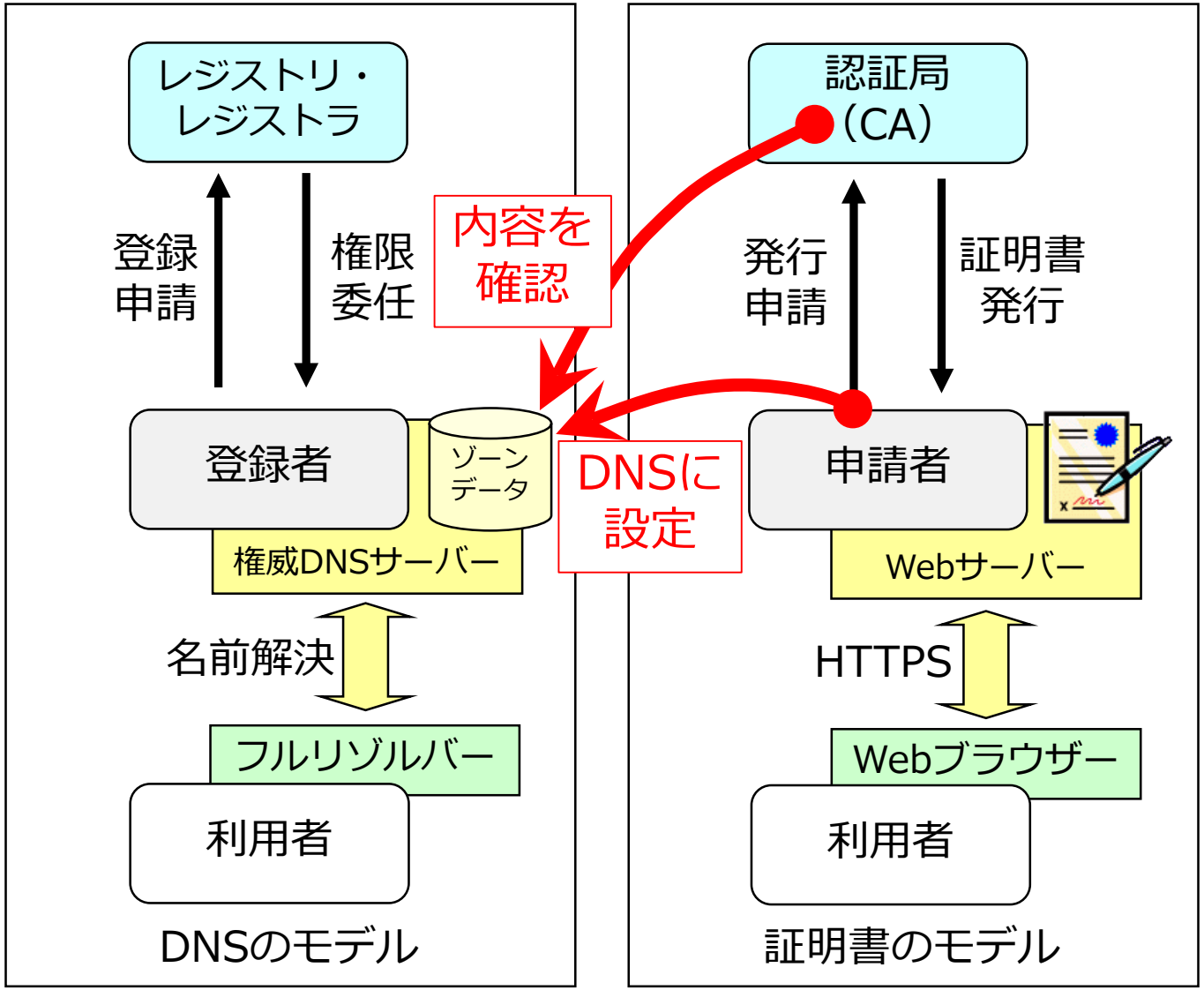
DNSのモデル



証明書のモデル

DNSと証明書―最近の関係

- 証明書の発行手続きにおいて、申請者からCAへの情報の伝達にDNSを使うケースが出て来ている
  - 申請者が発行可否情報や本人確認情報を、自身の権威DNSサーバーに設定
  - CAが設定内容を確認



パート2の内容について

- 申請者からCAへの情報の伝達にDNSを使うものの例として、パート2で以下の二つを解説
  - CAAレコード
  - 自動証明書管理環境（ACME）におけるDNS経由での認証
- 共に、DNSを用いた証明書関連技術の一つ
- 最近、これら二つの実装・普及が進み始めている

2. DNSを用いた証明書関連技術

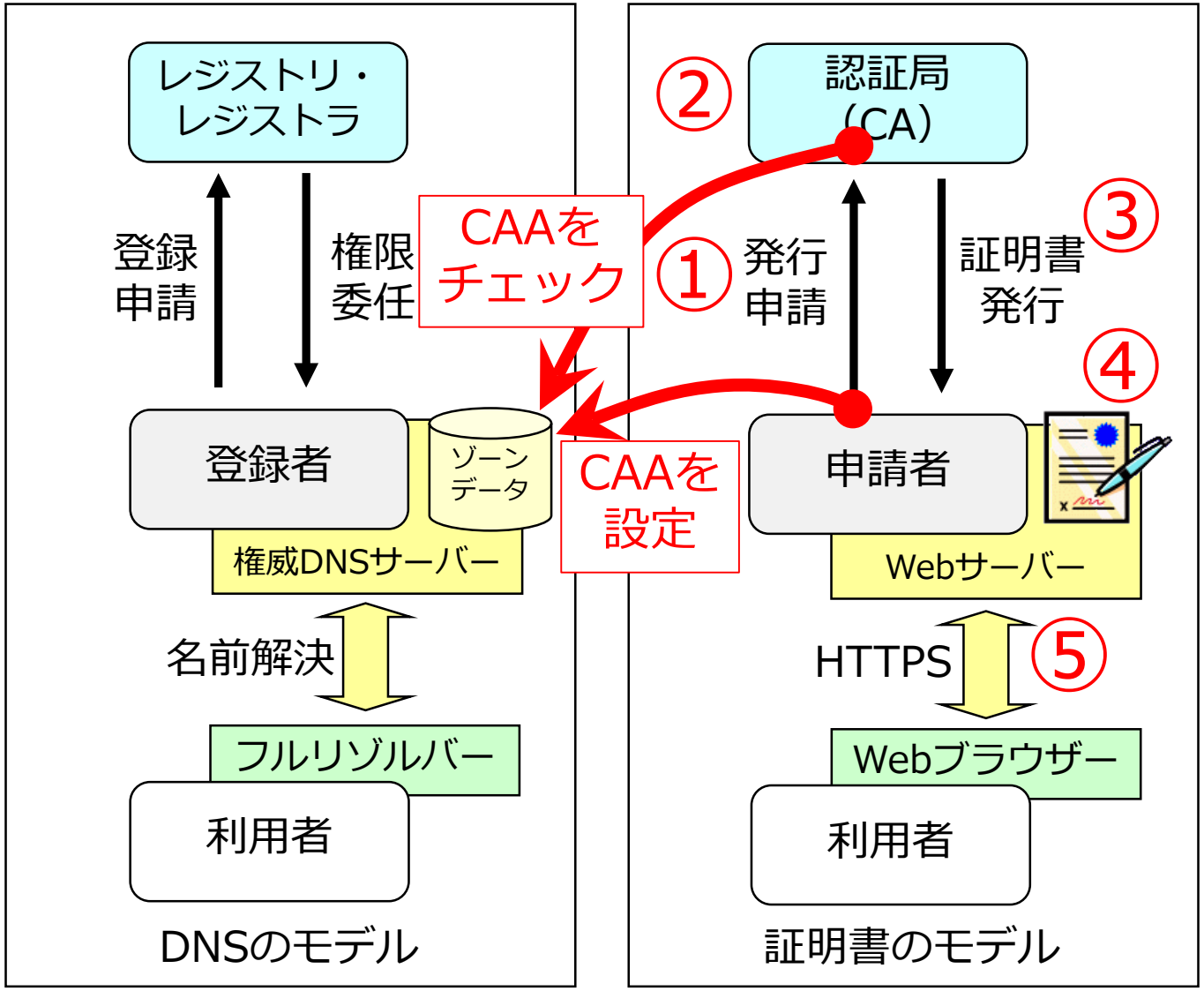
- CAAレコード
- ACMEにおけるDNS経由での認証

CAAレコードとは

- Certification Authority Authorization（認証局の許可）
- DNSのリソースレコードの一つ
  - A/AAAA、MX、TXTレコードなどと同様
- RFC 6844として、2013年に標準化
  - DNSではなく、PKIのWG（pkix WG）で標準化された

CAAレコードとは

- 証明書の発行申請に際し、申請者が自身の権威DNSサーバーに設定
- 証明書の発行手続きにおいて、CAが設定内容をチェック
- 右図の①と②の手順において、DNSを情報の伝達に利用
- DNSSECの利用を強く推奨



CAAレコードに設定される内容とその目的

- 内容：以下の2項目
  - 証明書の発行を許可するCA
  - 発行を許可しないCAに発行要求があった際の、連絡先と連絡手段
- 目的：証明書の発行における事故・トラブルの防止
  - 許可しないCAから、自身の証明書が発行されるのを防ぐ
  - 許可しないCAに、証明書発行要求があったことを知る

CAAレコードの設定は任意であり、設定がない場合は従来通り（発行制限なし）

CAAレコードの設定例とその意味

```
①  example.jp.  IN CAA 0 issue "jprs.jp"
①  example.jp.  IN CAA 0 issue "ca.example.com"
②  example.jp.  IN CAA 0 issuewild ";"
③  example.jp.  IN CAA 0 iodef "mailto:security@example.jp"
```

CAAレコードの設定例

① example.jpの証明書の発行を、「jprs.jp」と「ca.example.com」に許可
- 複数のCAに許可する場合、issue/issuewildをCAごとに指定
- CAの指定には、各CAが公開したドメイン名を設定

② example.jpのワイルドカード証明書の発行は、どのCAにも不許可
- 証明書の発行を禁止する場合、”;”を設定

③ 許可されていないCAが証明書の発行要求を受けた場合、<security@example.jp>に、電子メールを送ってほしい

CAAレコードによる判断の流れ

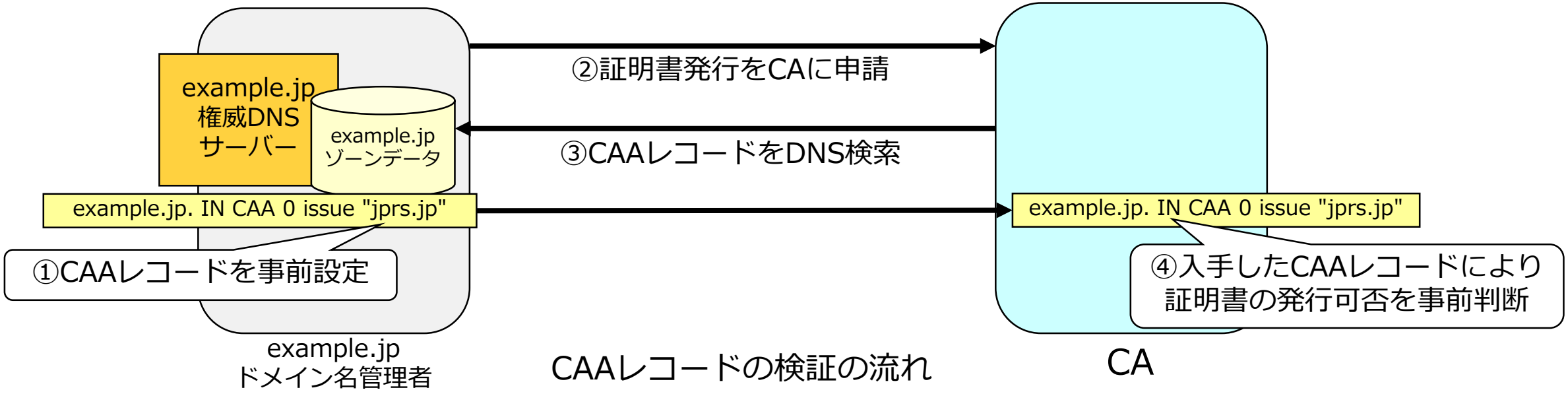
① CAAレコードを事前設定

② 証明書発行をCAに申請

③ CAがCAAレコードをDNS検索

④ 入手したCAAレコードにより、証明書の発行可否を判断

- 許可されていれば、以降の手順（審査、発行）へ



CAAレコードの検証の流れ

CAAレコードの検索における注意点

- CAAレコードが見つからない場合、TLDまでさかのぼって検索
  - RFC 6844で定義

例：www.example.co.jpのサーバー証明書を発行する場合の検索手順

① 「**www.example.co.jp**」のCAAレコードを検索 ⇒ 見つかった場合検索終了、見つからない場合②へ

② 「**example.co.jp**」のCAAレコードを検索 ⇒ 見つかった場合検索終了、見つからない場合③へ

③ 「**co.jp**」のCAAレコードを検索 ⇒ 見つかった場合検索終了、見つからない場合④へ

④ 「**jp**」のCAAレコードを検索 ⇒ 見つかった場合検索終了、見つからない場合⑤へ

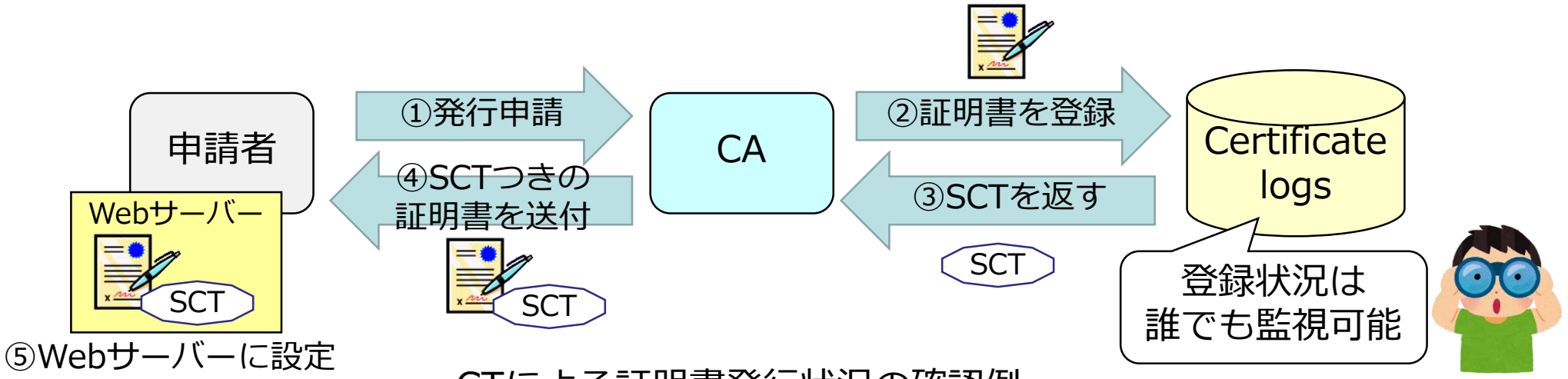
⑤ 検索終了、CAAレコードは設定されていなかったと判断

- 親ドメインのCAAレコードの設定により、予期しない形で証明書の発行が制限されてしまう場合がある

注：jpやco.jpなどにCAAレコードは設定していません

参考：CT（Certificate Transparency）との違い

- CAA: 証明書の誤発行を、発行前に予防・検知
- CT: 証明書の誤発行を、発行後に早期検知
  - CTは、証明書の発行状況をみんなで監視する仕組み



CTによる証明書発行状況の確認例

SCT: Signed Certificate Stamp（証明書データが格納されたことを示すタイムスタンプ情報）

CAAレコードのサポート状況

- 業界団体による検証の必須化（2017年9月8日以降）
  - CA/Browser Forumが、
    証明書発行時のCAにおけるCAAレコード検証を必須化

Ballot 187 - Make CAA Checking Mandatory - CAB Forum
<https://cabforum.org/2017/03/08/ballot-187-make-caa-checking-mandatory/>

  - 証明書が誤発行される事故が相次いだことが、その背景に存在
- 既に、証明書発行時に全CAがCAAレコードを検証している（はず）

DNSソフトウェアにおけるサポート状況

- CAAレコードの書式を標準サポート
  - BIND 9.9.6以降
  - NSD 4.0.1以降
  - PowerDNS Authoritative Server 4.0.0以降
  - Knot DNS 2.2.0以降
  - Windows Server 2016
- 書式をサポートしていない場合、RFC 3597の形式で記述可能

```
example.jp. IN TYPE257 ¥# 14 00056973737565 6A7072732E6A70
```

RFC 3597に基づいた記述例（上記は「example.jp. IN CAA 0 issue "jprs.jp"」と同じ内容）

DNSプロバイダーにおけるサポート状況

- CAAレコードの設定を標準サポート
  - Amazon Route 53
  - Dyn Managed DNS
  - Google Cloud DNS
  - Neustar UltraDNS
  - さくらインターネット ドメインメニュー
- ベータ版サポート（有効にする場合、プロバイダーに要連絡）
  - CloudFlare Global Managed DNS

IETFにおける問題点の指摘

- IETFのlamps WGでCAAレコードの仕様の問題点が指摘され、改定作業が進行中

lamps: Limited Additional Mechanisms for PKIX and SMIME
（PKIとS/MIMEへの限定的な機能追加を行うWG）

- 指摘された問題点
  - 「普及そのものが問題（Deployment = Issues）」
    - 現在の仕様が、CA/Browser Forumにより半強制的に普及することを問題視
  - CNAME/DNAMEを設定した場合の検索アルゴリズム
    - CNAMEを設定した場合の、CAAレコードの検索アルゴリズムの問題点が指摘
    - Prefixed nameを使うのがよいのではという提案あり
      - DNAMEではprefixed nameを設定できない点が指摘され、作業継続中

prefixed name: _*prefix*.example.jpのように、所定のラベルを前置したドメイン名

まとめ：CAAレコード

- 証明書の申請者が、自身の権威DNSサーバーに設定
  - 証明書の発行前に、CAが設定内容をチェック
- 証明書発行における、事故・トラブルの防止が目的
- 特殊な検索アルゴリズムに注意
  - CAAレコードが見つからない場合、TLDまでさかのぼって検索
- CA/Browser Forumが、CAAレコード検証を必須化
- IETFで仕様の問題点が指摘され、改定作業が進行中

自動証明書管理環境（ACME）とは

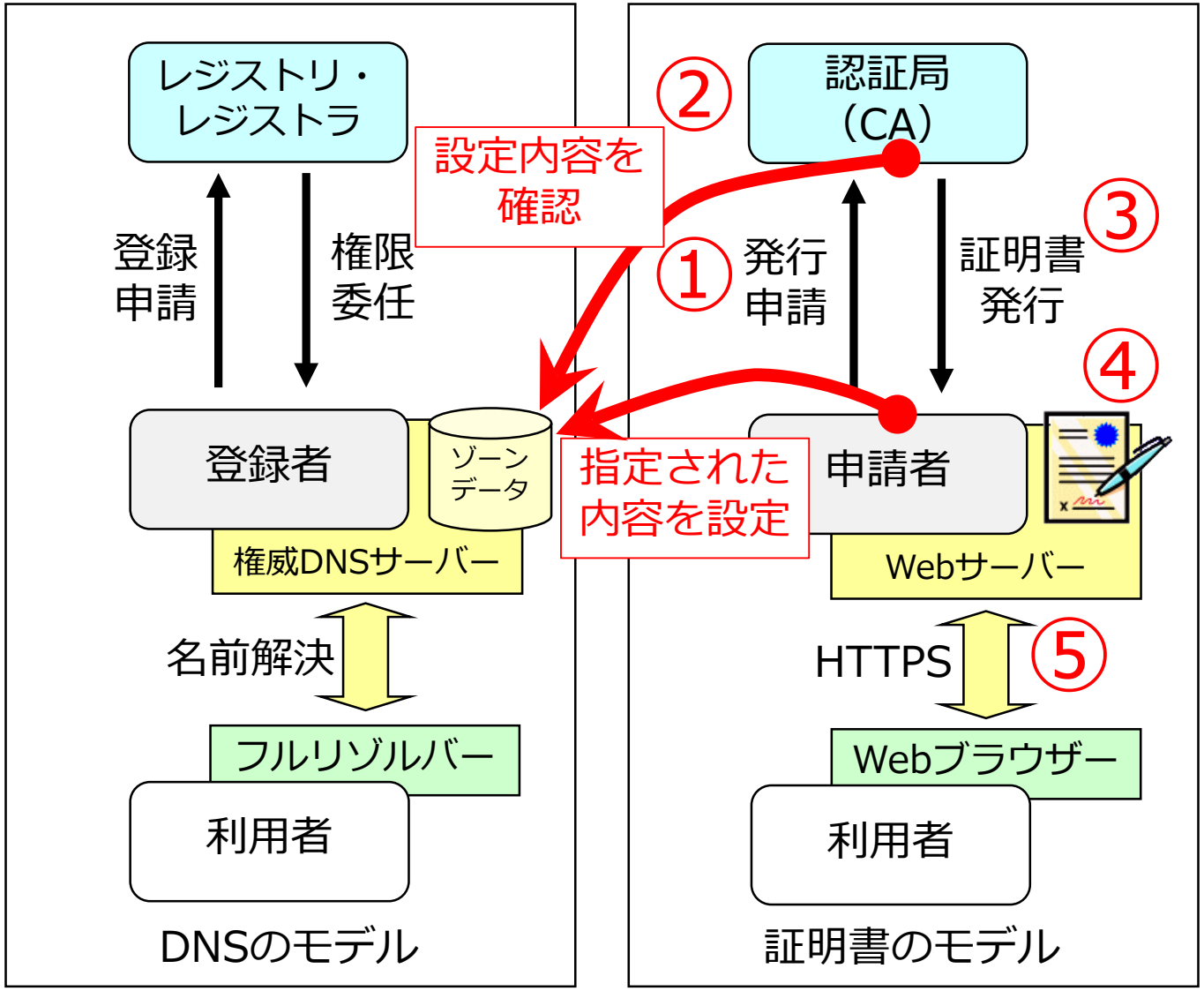
- Automatic Certificate Management Environment
- 証明書の管理を自動化するためのプロトコル
  - 検証・発行・失効など、一連のプロセスの自動化が目的
- IETF acme WGでの作業を完了し、IESGに送られた状態（2017年11月22日現在）

> Automatic Certificate Management Environment (ACME)
> <https://tools.ietf.org/html/draft-ietf-acme-acme-08>

- DNSを利用したバリデーションの方式として、dns-01を定義

dns-01とは

- 証明書の発行手続きにおける
  ドメイン名の管理権限の確認に、
  DNSを利用する方式
  - CAに指定された内容を、自身の権威DNSサーバーに設定
  - CAがそれを確認することで、申請者が管理権限を有していることを証明
- 右図の②の手順において、DNSを情報の伝達に利用
- DNSSECの利用を強く推奨



dns-01の設定例

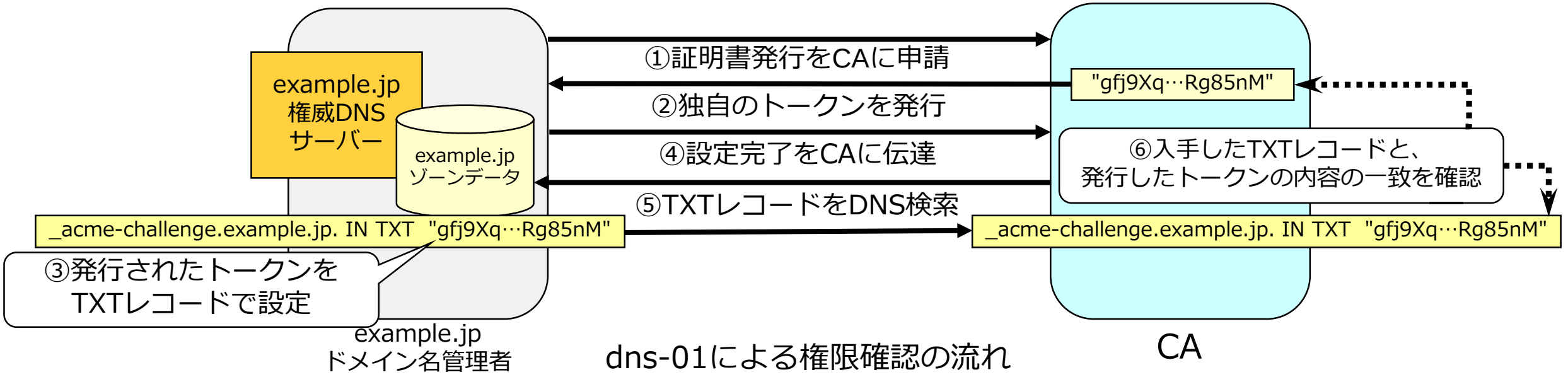
```
_acme-challenge.example.jp. IN TXT "gfj9Xq...Rg85nM"
```

dns-01の設定例

- _acme-challengeという、専用のprefixed nameを使用
  - *_acme-challenge*.example.jpのTXTレコードを設定できた場合、その管理者はexample.jpの管理権限を有していると判断
- CAに指定されたトークンを、TXTレコードで設定

dns-01による権限確認の流れ

① 証明書発行をCAに申請
② CAが独自のトークンを発行
③ トークンをTXTレコードで設定
④ トークンの設定完了をCAに伝達
⑤ CAがTXTレコードをDNS検索
⑥ CAがTXTレコードとトークンの内容一致を確認
- 確認できたら、証明書発行へ



dns-01による権限確認の流れ

共用DNSサービスにおける注意点

- Prefixつきのドメイン名とprefixなしのドメイン名の管理者が、同一であると想定
- 共用DNSサービスの運用形態により、問題が起こりうる
  - 例：攻撃者がprefixつきのドメイン名を同じDNSサービスに追加登録し、証明書の不正発行を図る
  - いわゆる親子同居問題として、JPRSが2012年に注意喚起を公開済

サービス運用上の問題に起因するドメイン名ハイジャックの危険性について
<https://jprs.jp/tech/security/2012-06-22-shared-authoritative-dns-server.html>

  - この問題はprefixed nameを使う、すべてのプロトコルに当てはまる

dns-01のサポート状況

- Let’s Encryptのサポートが先行
  - Let’s EncryptはACMEベースのCA実装、boulderを公開

GitHub - letsencrypt/boulder: An ACME-based CA, written in Go.
<https://github.com/letsencrypt/boulder>

- 独自方式の「DNS認証」をサポートするCAはいくつか存在
  - 設定対象のドメイン名や設定内容が、dns-01と異なる
  - 標準化の完了後、dns-01に変更するCAが増える可能性あり

まとめ：ACMEにおけるDNS経由での認証

- 証明書の申請者が、自身の権威DNSサーバーに設定
  - 証明書の発行時に、CAから指定された内容を設定
- 証明書発行における、ドメイン名の管理権限の確認が目的
- _acme-challengeという、専用のprefixed nameを使用
- 共用DNSサービスの運用形態に注意
- Let’s Encryptのサポートが先行

3. 最近の関係を踏まえ、DNS運用者がすべきこと

本パートで解説する項目

① ライフサイクルの一致

② リソースレコードタイプの増加

③ 標準化・意思決定による影響

④ 新たな注意点（はまりどころ）

⑤ DNSSECとの関係

①ライフサイクルの一致

- それぞれのライフサイクルを一致させる必要がある
  - ドメイン名のライフサイクル
  - 証明書のライフサイクル
- DNSを用いた証明書関連技術の出現により、
  ライフサイクル一致の重要性が、以前よりも更に増している

どう対応すべきか？

- 各組織における管理体制の確立
  - ドメイン名・DNSの管理と証明書の管理の連携
- 登録・廃止の際に必要な作業手順の確認と実行
  - 例：ドメイン名を廃止する場合、証明書も併せて失効する

②リソースレコードタイプの増加

- RFC 5507（Design Choices When Expanding the DNS）
  - 2009年発行、著者はIAB
  - 新しいデータをDNSに追加する場合の、拡張方法の比較・考察
    - リソースレコードタイプの追加を好ましい解決策（preferred solution）とし、TXTレコードの利用をほぼ確実に最悪（almost certainly the worst）としている
- 2010年以降、18種類のリソースレコードタイプが追加
  - 増加したリソースレコードタイプ（追加順）
    - HIP、TALINK、TLSA、NID、L32、L64、LP、EUI48、EUI64、**CAA**、CDS、CDNSKEY、CSYNC、URI、OPENPGPKEY、AVC、SMIMEA、DOA

今後もリソースレコードタイプの増加が見込まれる

どう対応すべきか？

- DNS運用者の視点
  - 新しいリソースレコードタイプの仕様・目的・内容の理解
  - 各組織における運用手順の検討・確立
  - 必要に応じたレコードの設定・運用
    - 権威DNSサーバーやフルリゾルバーのバージョンアップが必要になる場合あり
- DNSプロバイダーの視点
  - どのリソースレコードタイプのサポートを優先すべきかの判断
    - 以下の資料が参考になる

増え続けるRR Typeとどう付き合う？（IIJ 其田学氏：DNS Summer Day 2017）
<https://dnsops.jp/event/20170628/DSD2017_RRTYPE.pdf>

③標準化・意思決定による影響

- 標準化による影響
  - 例：ACME
    - IETF acme WGにおける作業が完了
    - 今後、IESGのレビューを経てRFCとなる予定
- 意思決定による影響
  - 例：CAAレコード
    - CA/Browser Forumでの意思決定
      - 証明書発行時の、CAにおけるCAAレコード検証必須化

IETFの標準化や業界の意思決定により、状況が変化

どう対応すべきか？

- 相手を知る
  - 主なステークホルダーは誰か？
  - それぞれのステークホルダーの考え（思惑）は何か？
  - 標準化や意思決定の場所・仕組みはどうなっているか？
- 動きを知る
  - Webブラウザーベンダーの動向
  - CAの動向
  - IETFにおける標準化の進捗動向
  - CA/Browser Forumのballot（投票）動向

Ballots - CAB Forum
<https://cabforum.org/ballots/>

④新たな注意点（はまりどころ）

- DNS運用・サービス提供における新たな注意点が存在
- 例1：CAAレコード
  - CAAレコードの検索アルゴリズム
    - CAAレコードが見つからない場合、TLDまでさかのぼって検索される
    - CNAME/DNAMEを設定した場合の、検索アルゴリズムの問題
- 例2：ACMEのdns-01認証
  - _で始まるprefixed nameの取り扱い

どう対応すべきか？

- 仕様の理解
  - はまりそうな部分はどこか？
- その必要があれば、運用でカバー
  - “A law is a law, however undesirable it may be”
    - 向こう（証明書関連のステークホルダー）もたぶん、そう思っている・・・
- 互いの理解と連携
  - Internet Week 2015のテーマ

    「手を取り合って、垣根を越えて。」
- 可能であれば、標準化活動への参加

⑤DNSSECとの関係

- CAAレコード・ACMEのdns-01認証の双方とも、DNSSECの利用を強く推奨
- 背景：DNSの信頼性が、証明書の信頼性に直接影響するようになった
- 証明書発行手続きの信頼性向上を図れる
  - DNSSECにより、データ出自の認証とデータの完全性を保証
    - 申請者が登録したデータであること
    - CAが受け取ったデータが書き換えられたり、失われたりしていないこと

改めて、DNSと証明書の現在の関係は？

- アドレスバーの中で、インターネットを一緒に支えている
- 担当する役割が違っていて、補完しあう関係にある
- どちらの役割も、インターネットにとって重要である
- そして・・・
  - **証明書の仕組みにも、DNSがより深くかかわるようになってきた**

互いがそれぞれをよく知り、うまく使うことで
「向き合っていく」ことが重要

おわりに：JPRSの技術情報発信

- JPRS DNS関連技術情報
  <https://jprs.jp/tech/>
- JPRS トピックス & コラム
  <https://jprs.jp/related-info/guide/>
  – Internet Weekの展示ブースでも配布
- メールマガジン「FROM JPRS」
  <https://jprs.jp/mail/>
- JPRS サーバー証明書発行サービス
  <https://JPRSサーバー証明書.jp/>

- JPRS 公式SNSアカウント


@JPRS_official


JPRSofficial

そして、Internet WeekのJPRSランチセミナーは今年で11年目

That’s it!