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～

2018年6月1日

Internet Week ショーケース in 広島

株式会社日本レジストリサービス（JPRS）

森下 泰宏

本資料はInternet Week 2017 ランチセミナー資料のUpdate版です

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
- 広島との関わり
  - 1992年: WIDE広島NOC設置
  - 1996年: IP Meeting 96 in 広島
  - 2009年: IETF 76 in 広島
  - 2018年: JANOG41 in 広島

これらをすべて
リアルタイムで経験しました

つまり、古い人ということです・・・

本日の内容

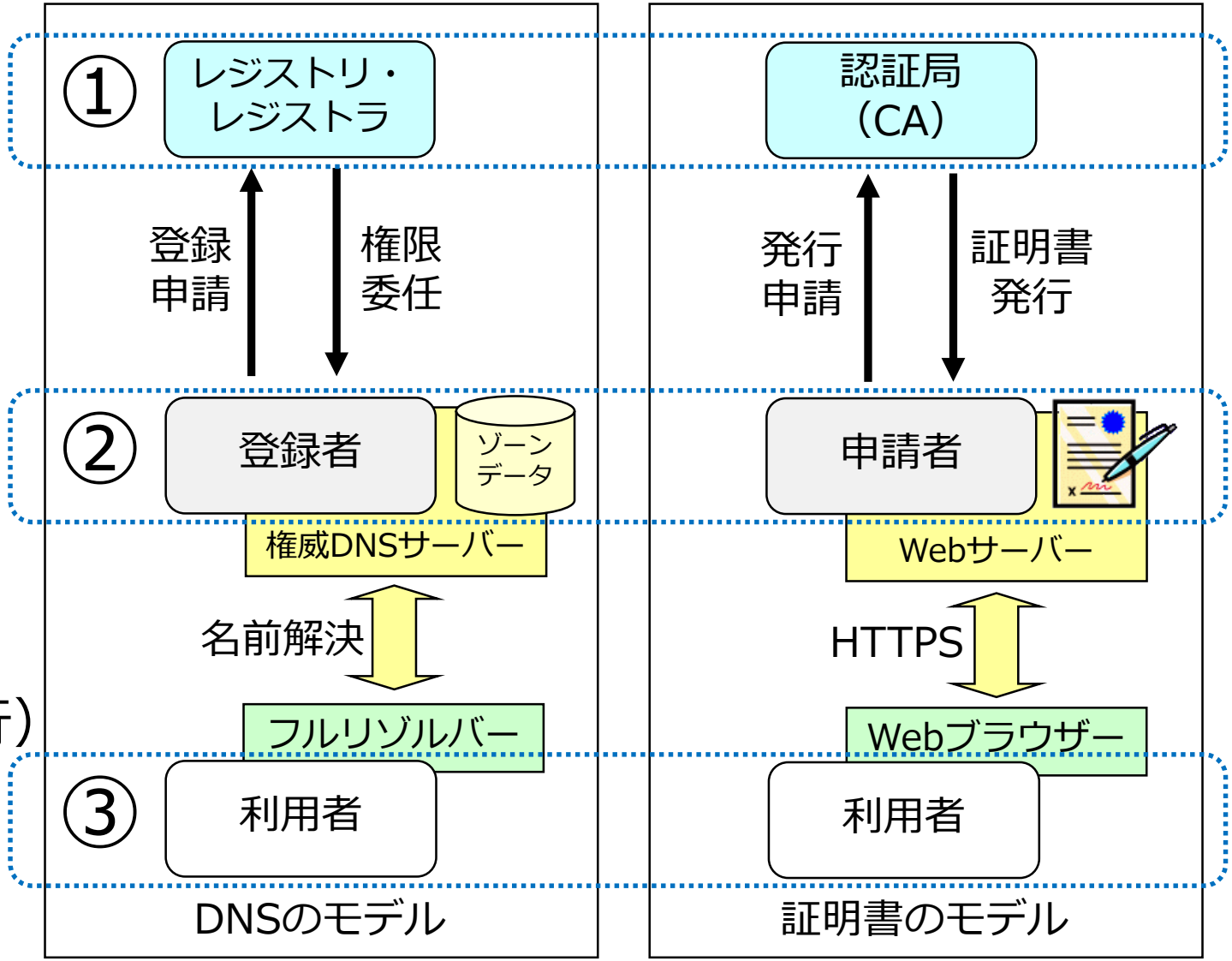
1. DNSと証明書の最近の関係
   - DNSと証明書の登録・発行モデルの類似性
   - DNSと証明書の関係・最近の関係
   - 認証局（CA）への情報伝達にDNSを使う例
     - CAAリソースレコード
     - 自動証明書管理環境（ACME）における、DNS経由での認証
2. 最近の関係を踏まえ、DNS運用者がすべきこと

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

1. DNSと証明書の最近の関係

DNSと証明書の登録・発行モデルの類似性

- 利用の形態
  - ① 提供する人
  - ② 登録・申請・設定する人
  - ③ 利用する人
- 申請から利用までの流れ
  1. 申請（登録申請・発行申請）
  2. 確認（所定の方法で審査）
  3. 提供（権限委任・証明書発行）
  4. 設定（サーバーに設定）
  5. 利用（名前解決、HTTPS）



DNSと証明書の（そもそもの）関係

- 証明書の発行（メール認証、ファイル認証）
  - メール認証：そのドメイン名の電子メールを受信可能
  - ファイル認証：そのドメイン名のWebサイトの中身を操作可能

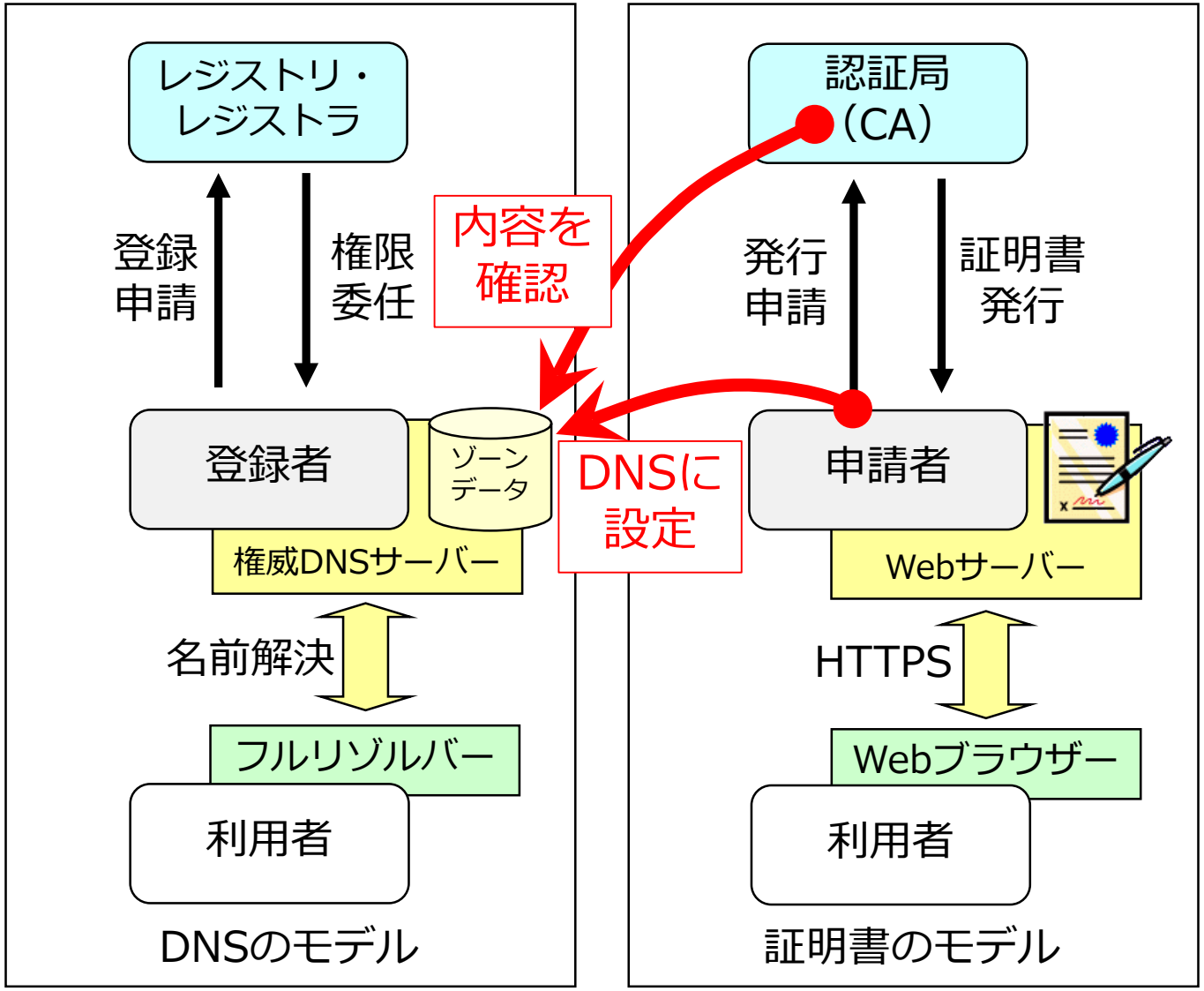
  ⇒ どちらも、そのドメイン名のDNS設定が正しいことが前提
- 証明書の利用（Webサイトへのアクセス）
  - DNS：相手に接続するための情報を提供
  - 証明書：接続した相手との安全な通信を実現

  ⇒ どちらが誤っていても、相手との安全な通信は不可能

DNSと証明書の最近の関係

- 証明書の発行手続きにおいて、申請者から認証局（CA）への情報伝達にDNSを使うケースが出て来ている
  - 申請者が証明書の発行可否情報やドメイン名の管理権限確認情報をDNSに設定
  - CAが設定内容を確認

従来からの「縦の関係」に加え、DNSと証明書の間を横断する、「横の関係」が出て来ている



認証局（CA）への情報伝達にDNSを使う例

- 本日は、以下の二つについて解説
  - CAAリソースレコード
  - 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

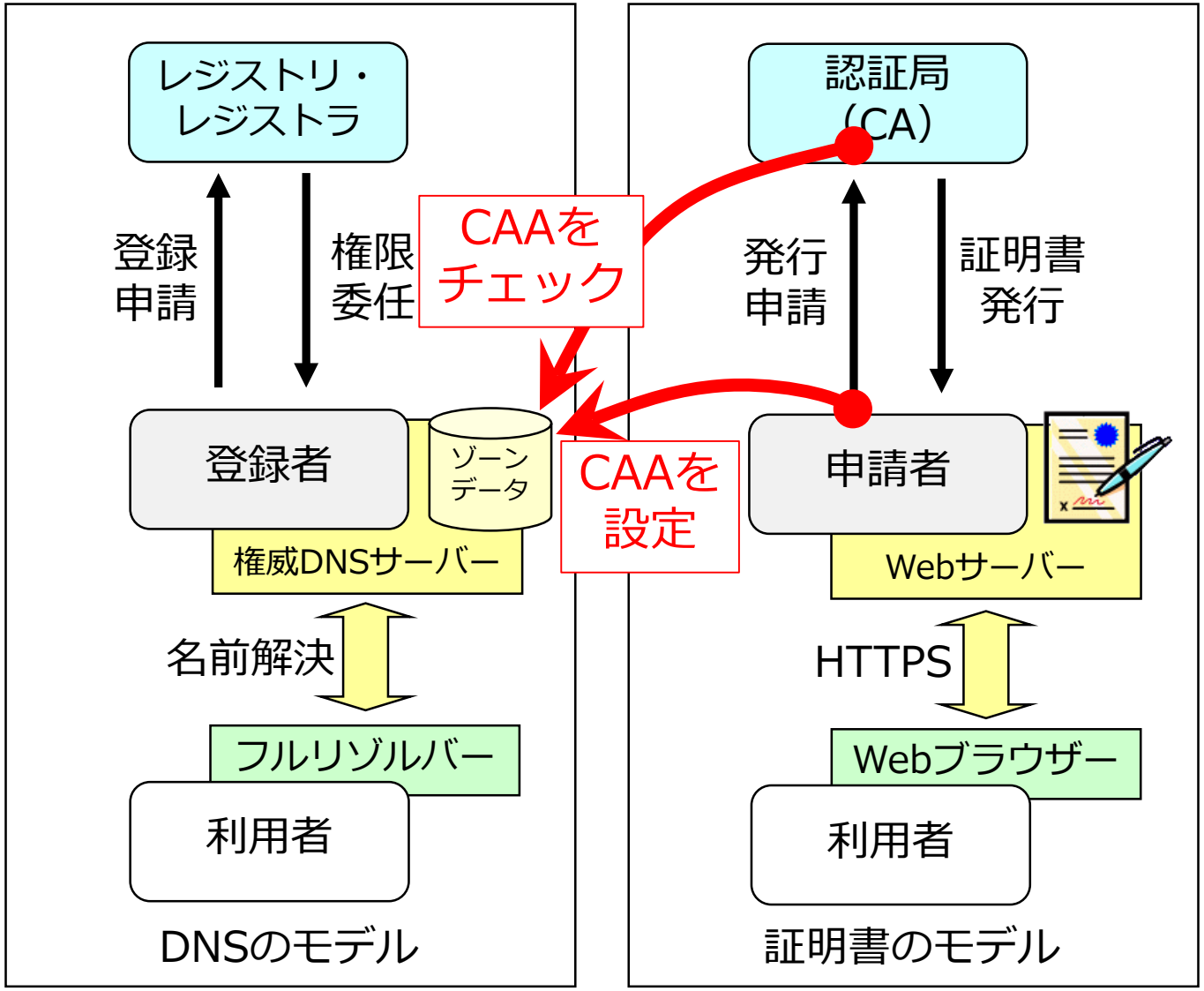
CAAリソースレコードとは

- 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リソース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

RFC 6844: DNS Certification Authority Authorization (CAA) Resource Record <https://www.rfc-editor.org/info/rfc6844>

CAAリソースレコードの仕組み

- 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CAAを設定
- 証明書の発行申請と発行確認において、DNSを利用
  - 証明書の発行手続きにおいて、CAがCAAの内容をチェック
- DNSSECの利用を強く推奨



CAAリソース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の設定は任意で、設定がない場合の動作は従来通り（発行制限なし）

CAAリソース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リソースレコードの設定例

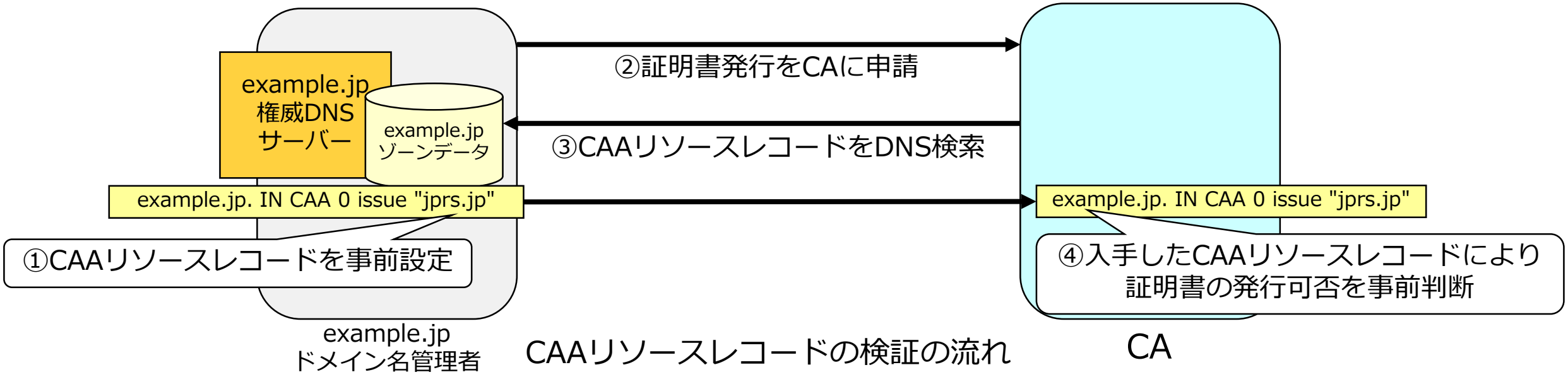
① example.jpの証明書の発行を、「jprs.jp」と「ca.example.com」に許可
- 複数のCAに許可する場合、issue/issuewildをCAごとに指定
- CAの指定には、各CAが公開したドメイン名を設定

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リソースレコードによる判断の流れ

① CAAリソースレコードを事前設定
② 証明書発行をCAに申請
③ CAがCAAリソースレコードをDNS検索
④ 入手したCAAリソースレコードにより、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リソースレコードの検索における注意点

- CAAリソース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

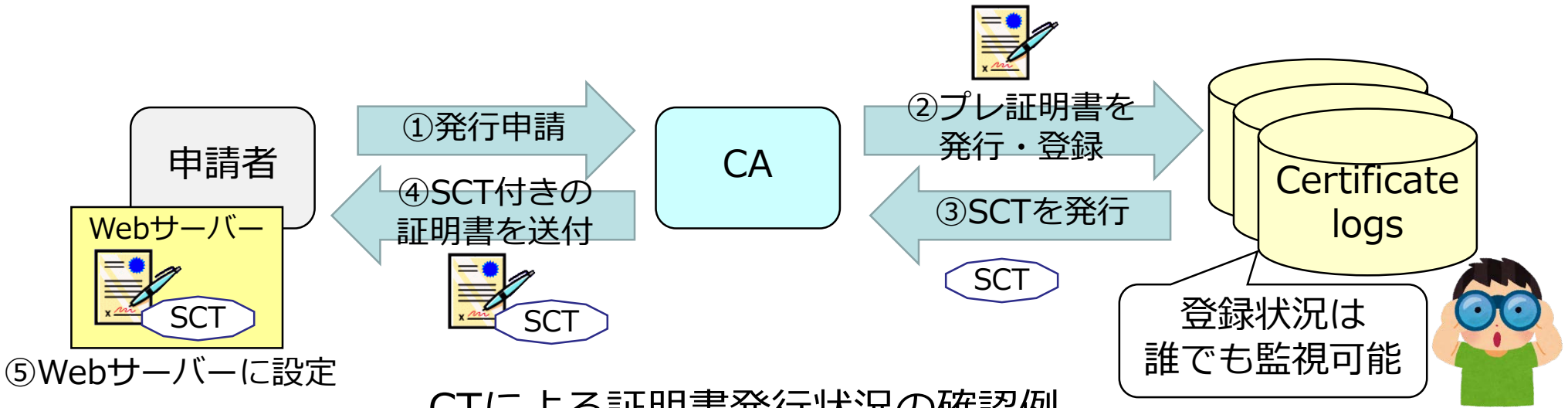
⑤ 検索終了、CAAリソースレコードは設定されていなかったと判断

- 親ドメインのCAAリソース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：JPRSではjpやco.jpなどにCAAリソースレコードを設定していません

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 Timestamp（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リソース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リソース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リソース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リソース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リソースレコードの設定を標準サポート
  - Amazon Route 53
  - Cloudflare Global Managed DNS
  - 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

まとめ：CAAリソースレコード

- 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リソース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、CAのCAAリソースレコード検証を必須化

自動証明書管理環境（ACME）とは

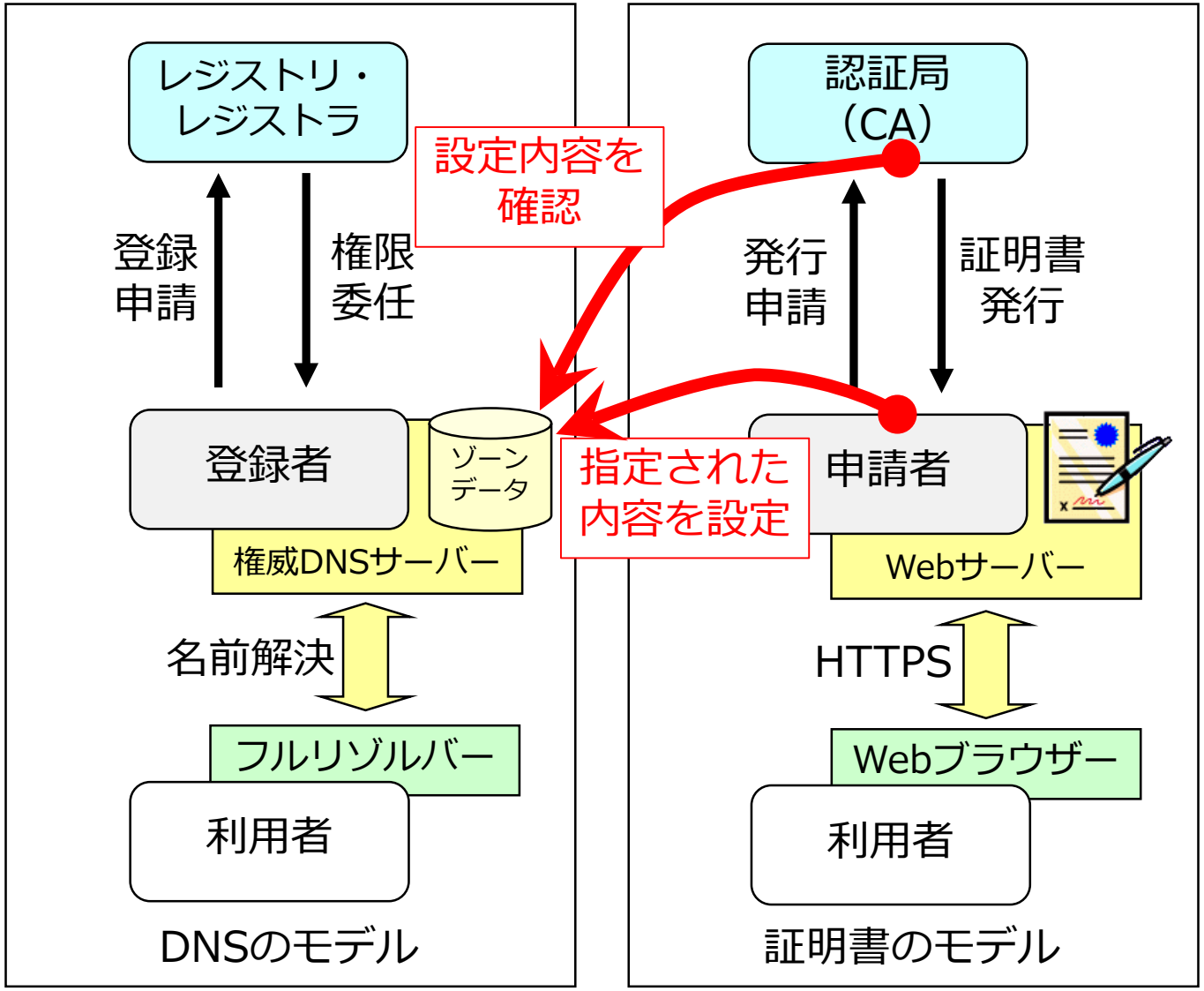
- Automatic Certificate Management Environment
- 証明書の管理を自動化するためのプロトコル
  - 検証・発行・失効など、一連のプロセスの自動化が目的
- IETF acme WGでの作業を完了、IESGでレビュー中
  （2018年5月16日現在）

Automatic Certificate Management Environment (ACME)
<https://tools.ietf.org/html/draft-ietf-acme-acme-12>

- 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
- 証明書の発行確認において、
  DNSを利用
  - 自身の権威DNSサーバーに、
    CAに指定された内容を設定
  - CAがDNS検索で設定内容を
    確認し、申請者が管理権限を
    有していることを検証
- DNSSECの利用を強く推奨



dns-01の設定例

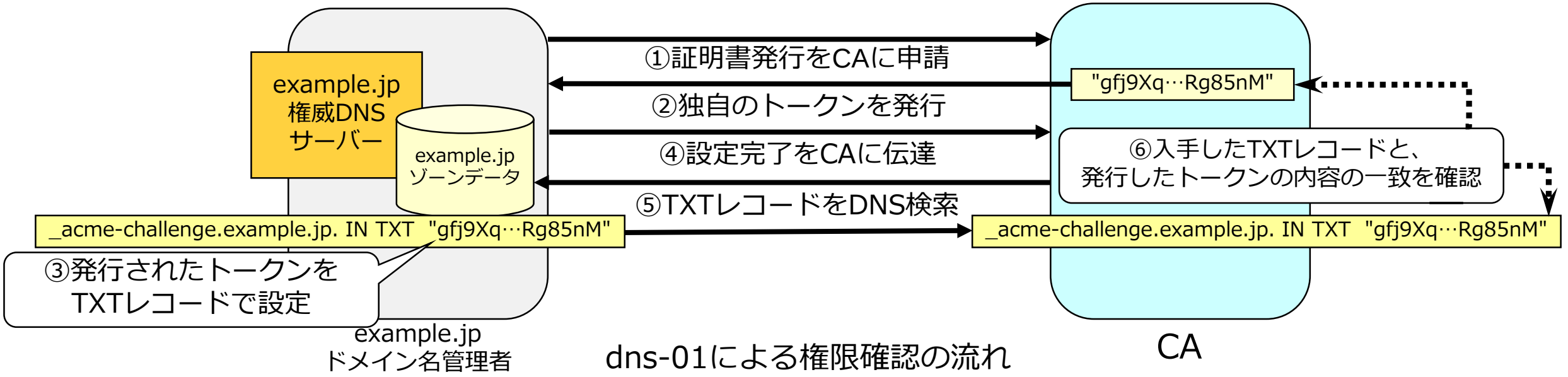
```
_acme-challenge.example.jp. IN TXT "gfj9Xq...Rg85nM"
```

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

dns-01のサポート状況

- Let’s Encryptのサポートが先行
  - 2018年3月から発行を開始したワイルドカード証明書では、dns-01が必須化されている
- 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

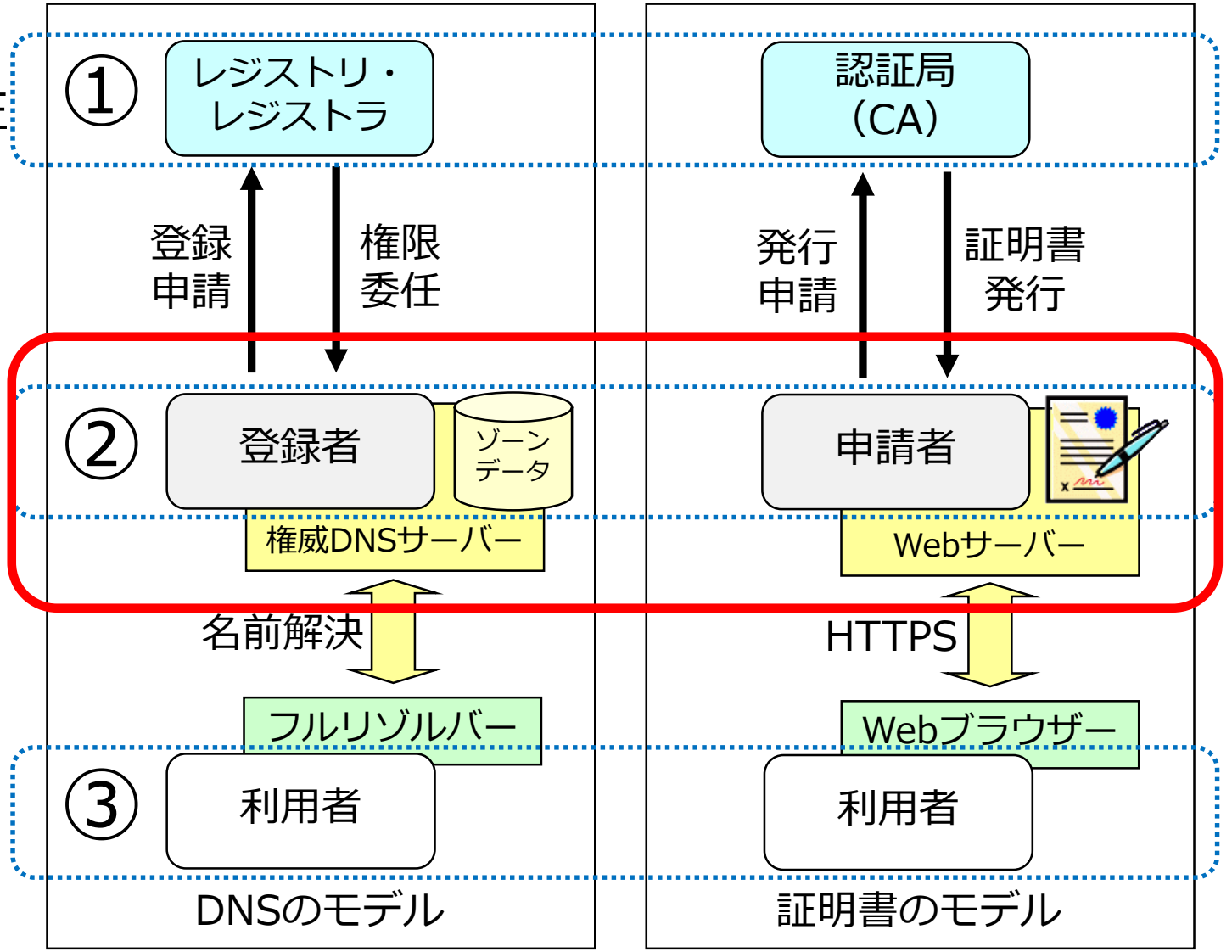
2. 最近の関係を踏まえ、DNS運用者がすべきこと

本パートで解説する項目

1. 管理における整合性の確保
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. 管理における整合性の確保

- ②登録・申請・設定における整合性
  - ドメイン名登録者と証明書申請者
  - 権威DNSサーバーとWebサーバー
- 整合性の確保が必要な項目の例
  - Webサイトで公開するドメイン名と、証明書のCNやSANsに指定するドメイン名
  - CAAリソースレコードの設定内容と、証明書を申請する認証局
  - そのドメイン名でサービス（Webサイト）を公開・継続する期間



どう対応すべきか？（1/2）

- ドメイン名・DNSの管理と証明書の管理の一元化・連携
  - 特に、担当部門が異なる場合の管理の連携
    - 例：企業など、ドメイン名・証明書は知財部門、DNS・Webサーバーはシステム部門がそれぞれ管理するといった形態がある
  - 外部のサービスや業者を使う場合の適切な管理・手順化
    - 例：DNSのリソースレコードや証明書の設定・更新方法の確認・手順化
    - 例：業者への委託、委託業者の変更における適切な情報管理・引き継ぎ

どう対応すべきか？（2/2）

- ドメイン名と証明書のライフサイクルの同期
  - ドメイン名の更新期限切れ、証明書の有効期間満了に注意
    - 組織内における登録・申請・更新手順の確認・明確化
  - サービス継続のためのコストの確保、体制・仕組み作り
    - 一度始めたサービス（Webサイト）を廃止することはリスク
  - やむを得ずサービスを廃止する場合の対応
    - レジストリに登録したネームサーバー設定の削除
    - 証明書の失効

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リソースレコード
    - CA/Browser Forumでの意思決定
      - 証明書発行時の、CAにおけるCAAリソース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リソースレコード
  - CAAリソースレコードの検索アルゴリズム
    - CAAリソースレコードが見つからない場合、TLDまでさかのぼって検索
    - 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

5. DNSSECとの関係

- CAAリソースレコード・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おり、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ではさまざまなチャネルで、
  ドメイン名・DNS・サーバー証明書に関する技術情報を発信中

＜情報発信の例＞

- 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/>
  – 各種イベントの展示ブースでも配布
- JPRS 公式SNSアカウント

- メールマガジン「FROM JPRS」
  <https://jprs.jp/mail/>
- サーバー証明書発行サービス
  <https://jprs.jp/pubcert/>

@JPRS_official

JPRSofficial

That’s it!