Originator Profile 対応

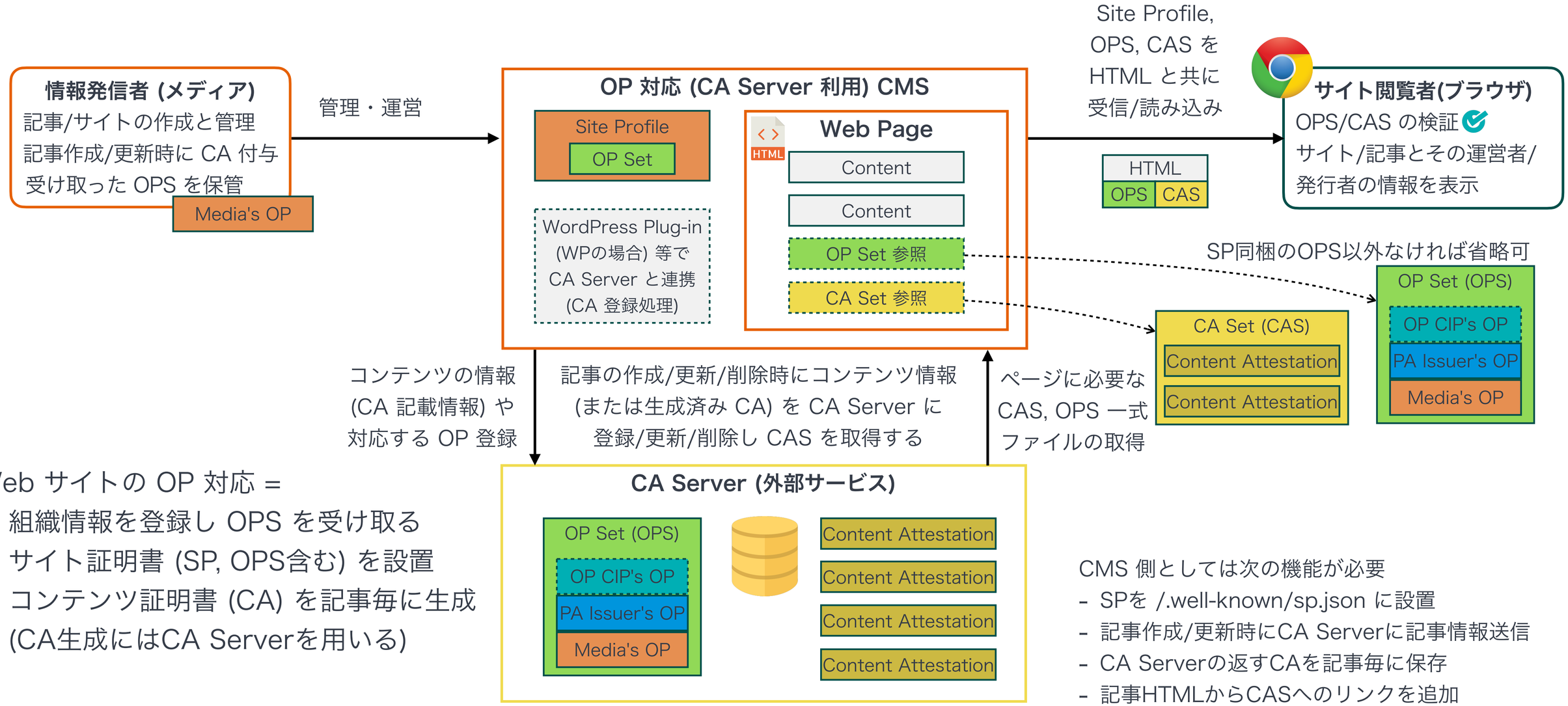
Originator Profile 対応

既存の Web サイトを OP 対応にするために必要なこと

- **OP 登録と OPS の受け取り** (事前に必要な手続き)
  1. OP 登録サイトのアカウントを作成、組織情報などを記載して OP 登録を申請
  2. OP CIP が記載内容を確認し OP ID と Originator Profile (OP) を発行
  3. 合わせて PA Issuer (第三者認証機関など) の OP なども含む OP Set (OPS) も提供
- **サイト証明書の発行** (サイト毎に実施、OPS 更新時には差し換えが必要)
  1. サイトの情報を記載した **Site Profile (SP)** を作成し .well-known/sp.json に設置
- **コンテンツ証明書の発行** (記事/コンテンツの作成/更新毎の自動処理が必要)
  1. 記事の作成/更新時に記事情報を記載した **Content Attestation (CA)** を作成
  2. ページ中の CA リストファイル CA Set (CAS) を作成し HTML に埋め込みまたはリンク

記事作成/更新時の処理は WordPress を例とした参照実装を CIP 側で用意/更新

Originator Profile の対応 (CA Server 利用)



Web サイトの OP 対応 =

1. 組織情報を登録し OPS を受け取る
2. サイト証明書 (SP, OPS含む) を設置
3. コンテンツ証明書 (CA) を記事毎に生成
   (CA生成にはCA Serverを用いる)

CMS 側としては次の機能が必要

- SPを /.well-known/sp.json に設置
- 記事作成/更新時にCA Serverに記事情報送信
- CA Serverの返すCAを記事毎に保存
- 記事HTMLからCASへのリンクを追加

処理の流れについて詳しくは「OPの仕組み」セクションを参照

CA Server の役割

署名鍵管理、CA の作成・署名、SP (WSP)の作成・署名

- OP 所有者の鍵ペアの作成と署名鍵の管理
  - CA (など) への署名に利用する鍵ペアを作成、署名鍵を安全に保管
  - (OP 登録時には CA Server で生成した公開鍵を登録し OP に含められます)
- CA の作成・署名
  - 未署名の CA 記載情報(コンテンツ情報)を受け取り CA (W3C VC 2.0 形式) を作成
  - CA 記載情報の補完: CA 対象コンテンツのハッシュ値を計算し補完
    - ハッシュ値計算は CMS 側で行うか CA Server 側で計算・補完させるかいずれも可能
- SP の作成・署名
  - 未署名の WSP 記載情報(サイト情報)を受け取り WSP (W3C VC 2.0 形式) を作成
  - あるいは WSP 記載情報とともに OPS を受け取り SP を作成
    - この場合は生成する SP の検証に必要な OP が OPS に含まれているの検証も行う

要はOP対応において管理・実装・処理が複雑な部分を可能な限り引き受ける

Originator Profile 対応の進め方

既存 Web サイトを OP 利用サイトにするためには

- 対応/実装方針の検討 (事前検討)
  - 手動での実験的対応か、CMS を改修/実装するか、CA Server を利用するか...
- OP 登録 (事前手続き)
  - 組織情報や保有証明書を OP CIP に登録し、OP (OPS) の提供を受ける
- サイトプロファイルの設置 (サイト毎に一度)
  - サイト(ドメイン)毎のサイトプロファイル (SP) を作成し .well-known 配下に設置
- コンテンツ証明書の発行 (記事/コンテンツ毎に一度)
  - 記事/コンテンツに対応するコンテンツ証明書 (CA) を作成し CAS, OPS にリンク

https://cip.docs.originator-profile.org/studies/general-instruction/

OP 対応: 対応/実装方針の検討

仕組みとイメージの確認だけなら手動対応での試験も可能

A. 手動で仕組みの確認をする実験 (最初の一歩として)

- 対応結果や仕組みの確認までであれば、CMS のプログラム改修は不要
- 各種証明書となる JSON ファイルの作成/設置とその参照用タグの挿入を手動で実施
- 各種証明書の JSON ファイルは CA Server (または CLI) で作成します (手順は後述)

B. CMS として対応する場合 (推奨)

- CA Server を利用する場合 (推奨)
  - 鍵管理と署名やハッシュ値計算を含む CA 生成処理をすべて CA Server に任せる
  - WordPress 利用サイト向けには WP Plugin と CA Server の参照実装あり
- すべて自前で実装する場合
  - 参照実装(Node パッケージや CLI 等)を利用または参考に CMS 側で個別に実装

CA Serverを利用しないケースについての開発支援については要相談

OP 対応: OP 登録

情報発信者としての組織情報を登録し OP の発行を受ける

- OP 登録手続き (事前手続き)
  1. OP 登録用の OP CIP サイトでの申請 (組織情報を登録し審査を受ける)
  2. OP CIP が記載内容を確認し OP ID と Originator Profile (OP) を発行
  3. 合わせて PA Issuer の OP なども含む OPS (OP Set) も提供
- 補足: 鍵ペアの作成と CA Server の利用について
  - OP 登録時の鍵ペア作成と検証鍵(公開鍵)提出は CA Server 利用時は不要
  - CA Server 側で鍵ペア(署名鍵/検証鍵)の生成/管理を行う仕組みです
- 補足: 現時点では OP と PA の登録・発行を共通システムで行いますが、将来的には OP 登録と並行して PA Issuer 毎のシステムにも登録と PA 発行の手続きが必要となる可能性があります

25年度より電通総研様のシステムで実施いただきます(マニュアルなどは別途)

OP 対応: サイトプロファイル(SP)の設置

サイトの情報に署名したサイトプロファイルをサイト毎に作成・設置

- サイトプロファイルの作成と設置
  1. サイトの情報を記載した JSON ファイル (未署名WSPデータ) を用意
  2. CA Server の REST API (あるいは CLI など) で署名した WSP を作成
  3. WSP と OPS をまとめたサイトプロファイル (SP, JSON ファイル)を作成
  4. サイトプロファイルを /.well-known/sp.json に設置
- 補足: OPS には有効期限があり SP は期限切れ前に更新作業が必要になります
- 詳細: https://cip.docs.originator-profile.org/studies/general-instruction/sp-setup-guide/

CA ServerではWSPもしくはSPを生成でき、後者の場合は手順3は省略

OP 対応: コンテンツ証明書(CA)の発行

記事やコンテンツとその情報に署名した証明書を記事毎に作成・設置

- 記事/コンテンツの証明書 (CA: Content Attestation) の作成と設置
  1. CA を発行する記事の対象範囲を決める (CSS Selector などで指定)
  2. コンテンツ情報を記載した JSON ファイル (未署名CAデータ)を用意
  3. CA Server の REST API (あるいは CLI など) で署名した CA を作成
  4. ページ中の CA をすべてまとめた CAS (CA Set, JSON ファイル) を作成
  5. ページ HTML 中に CAS (と OPS) を参照する script タグを挿入

- 詳細: https://cip.docs.originator-profile.org/studies/general-instruction/ca-target/
  https://cip.docs.originator-profile.org/studies/general-instruction/cas-setup-guide/

OP 対応: WordPress Pluginの利用

コンテンツ証明書(CA)の自動発行はプラグインの導入のみで可能

- 記事/コンテンツの証明書 (CA: Content Attestation) の作成と設置
  1. WordPress に CA 発行機能プラグイン (CA Manager) をインストール
  2. 電通総研様の CA Server との認証に必要なファイル設置や固定ページ作成
  3. 認証情報や(テーマなどに応じた)記事の対象範囲などをプラグインの設定画面に入力
  4. 記事や固定ページを更新 (プラグインが CA を自動作成・保存します)
  5. 結果を確認 (設定や他のプラグインとの相性で適切な CA 発行がされない場合あり)
- 補足: プラグイン導入後に公開・更新した記事にのみ CA が付与されるため、過去個記事への発行や CA の再発行をする場合は対象記事の更新が必要です
- 詳細: https://cip.docs.originator-profile.org/studies/general-instruction/wordpress/

プラグインの相性問題などを見つけた場合は共有頂けますと幸いです

OP 対応に関するドキュメント

CMS の OP 対応を設計・検討頂く際の参考資料

- OP 対応実験一般の案内
  - https://cip.docs.originator-profile.org/studies/general-instruction/
- サイト (サイト情報) の OP 対応 (Site Profile の作成と設置)
  - https://cip.docs.originator-profile.org/studies/general-instruction/sp-setup-guide/
- 各コンテンツの OP 対応 (CA の作成と設置)
  - https://cip.docs.originator-profile.org/studies/general-instruction/cas-setup-guide/
- WordPress プラグインの設定手順
  - https://cip.docs.originator-profile.org/studies/general-instruction/wordpress/
- CA における検証対象コンテンツ・HTML 要素について
  - https://cip.docs.originator-profile.org/studies/general-instruction/ca-target/

Appendix
ドキュメントと開発リソース

ドキュメントと実装ソースコード

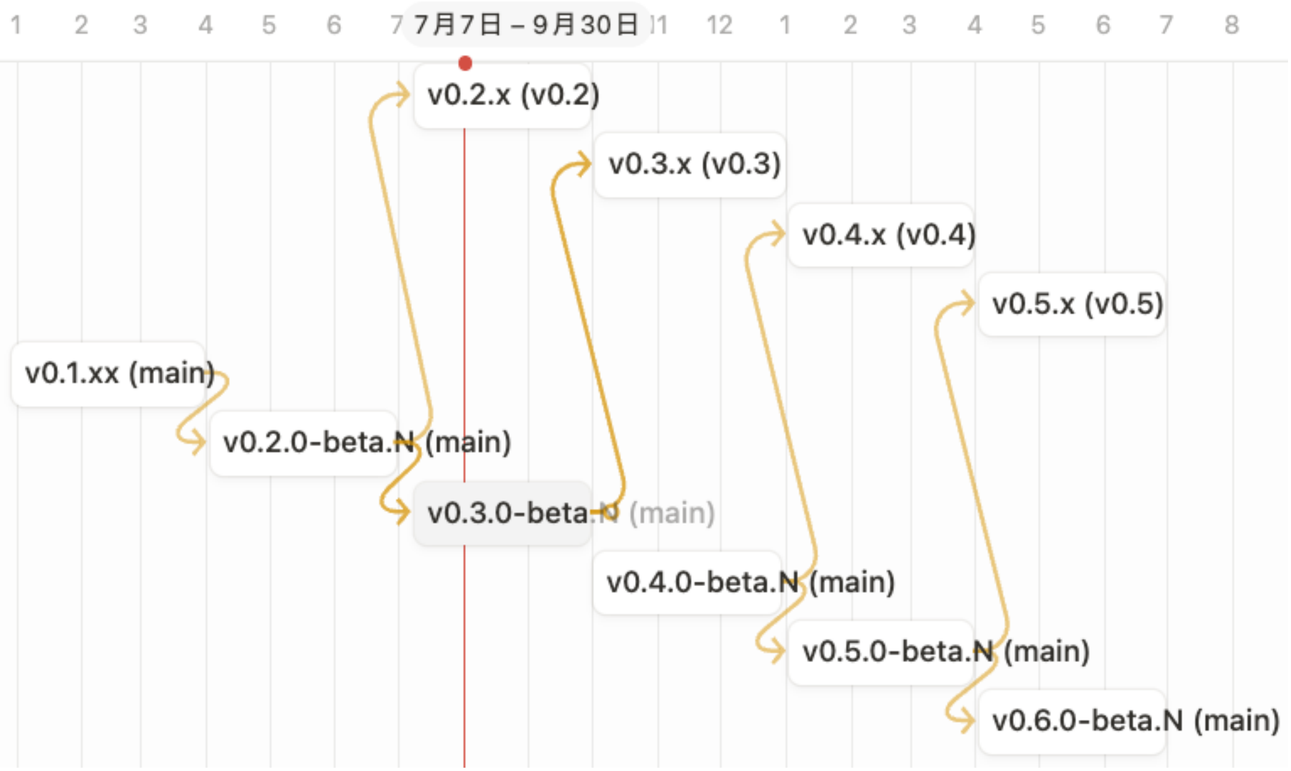
技術詳細説明や参照実装はこちらをご覧ください

- 技術説明ドキュメント
  - https://cip.docs.originator-profile.org/
- 実証実験案内 (昨年度分、今年度用は準備中)
  https://cip.docs.originator-profile.org/studies/
- Originator Profile Blueprint (仕様草案)
  - https://docs.originator-profile.org/opb/
- 参照実装ソースコード
  - https://github.com/originator-profile/profile-share
  - https://github.com/originator-profile/ca-server

開発支援ツールやライブラリなどの詳細はこれらのサイト内を参照してください

参照実装系の更新スケジュール＝3ヶ月サイクル

Beta/Release 型のスケジュールベースサイクルでリリース



- 参照実装系は四半期毎のリリースを行います
  - **3ヶ月毎の上旬リリース** (1日リリースではない)
  - 拡張機能、WordPress Plug-in、CA Server など
  - 参照実装系以外 (他のCMS対応、本番系のCA Server など)の更新スケジュールはこれに続けて定期更新される場合もあるがそれぞれ独立です
- 本番環境向けには v0.x ブランチをご利用ください
  - 基本的にはセキュリティ修正など最小限に
  - 実験都合などで小さな(後方互換の)修正などはあり得る
  - 3 ヶ月以上の継続サポートは必要に応じて検討
  - mainブランチは最新開発とベータリリース用であり、非互換変更やバグがあっても構わない方向けです

リリース: https://github.com/originator-profile/profile-share/releases