デジタルサイネージ標準システム相互運用ガイドライン第 2 版

【別紙 1 】

SPF インタフェース仕様書

第 1 版

2017 年 5 月

デジタルサイネージコンソーシアム

目次

1. はじめに

本書は 、デジタルサイネージコンソーシアム公表の「デジタルサイネージ標準システム相互運用ガイドライン（2.0 版）」（以降、ガイドライン）における「サイネージプラットフォーム」（以降本システム）のインタフェース仕様書である。

「サイネージプラットフォーム」とは、情報の一斉配信を実現するもので、複数の異なる各ベンダのサイネージ配信システムに対する一斉情報配信を実現するための基盤システムであり、各サイネージ配信システムと配下で接続することで、平時・有事の情報伝達において多数のサイネージへの相互接続性・運用性の確保することを目的とする。

本書は、情報の一斉配信において、各ベンダのサイネージ配信システムが本システムに接続するためのインタフェース仕様について規定したものである。

1.1. 処理シーケンス

本システムがサイネージ配信システムへコンテンツ配信する際の処理シーケンスを以下に示す。

処理シーケンスは大きく分けて 以下の 3 つのフェーズから構成される。

(1) 事前準備
- 各システムを接続する。

(2) 配信準備
- 情報提供者から一斉配信する情報が送信される。
- 情報を元にコンテンツを生成する。
  （HTML5 コンテンツや画像コンテンツ）

(3) 配信
- サイネージプラットフォームからサイネージ配信システムにコンテンツを配信する。
- サイネージ配信システムがコンテンツの配信結果をサイネージプラットフォームに報告する。

(4) サイネージ端末配信
- サイネージ配信システムからサイネージ端末にコンテンツを配信する。
  ※この処理は各デジタルサイネージシステムの範疇の為、本仕様書では定義しない。

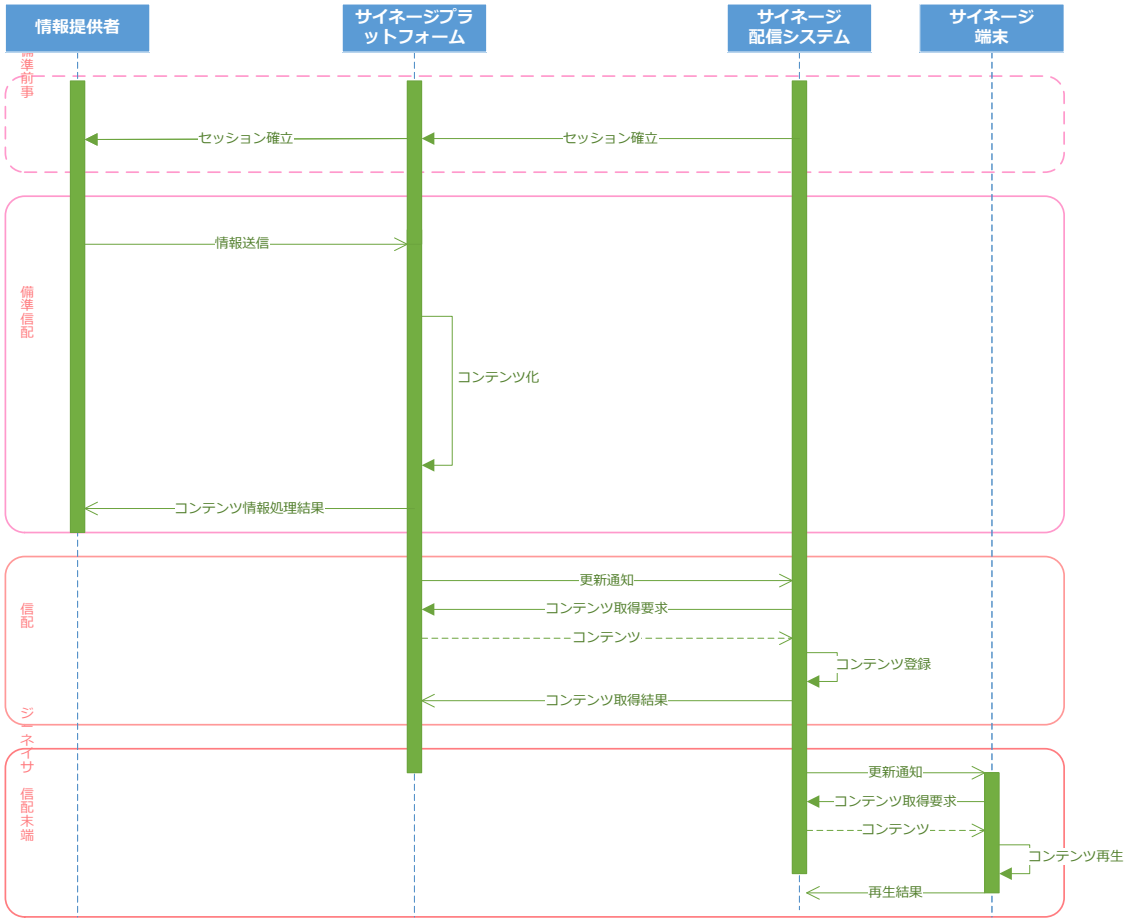

図 1　システム全体の処理シーケンス図

2. 通信フォーマット

サイネージプラットフォームと情報提供者間、およびサイネージプラットフォームとサイネージ配信システム間は、SPF-XML(以降「SPF-XML」と呼称)を活用し通信する。

2.1. SPF-XML フォーマット

別紙 2「SPF-XML 仕様書」を参照のこと。

3. SPF-S インタフェース仕様

SPF-S インタフェースとは、サイネージ配信システム向けのインタフェースであり、サイネージ配信システムからの情報を受信するためのものである。サイネージプラットフォーム・サイネージ配信システム間の通信は、サイネージ配信システムがサイネージプラットフォームにリクエストする形を想定し、サイネージプラットフォームが接続に必要なエンドポイントを提供する。

また更新通知インタフェースは、各社サイネージ配信システムのネットワーク・環境要件等を考慮し、PUSH 型と PULL 型を定義する。

表 1　通信インタフェース一覧

名称	プロトコル	メソッド	URL
セッション確立	ws (wss)	–	~/notification
更新通知（PUSH 型）	ws (wss)	–	–
更新通知（PULL 型）	http (https)	POST	~/notificationRequest
コンテンツ取得要求	http (https)	GET	該当コンテンツの URL
コンテンツ取得結果送信	http (https)	POST	~/sendResult

3.1. セッション確立

サイネージ配信システムが、サイネージプラットフォームに対し、更新通知を PUSH 型で受けるために必要なネゴシエーションを行うインタフェース。プロトコルは WebSocket を使用する。

当インタフェースでサイネージプラットフォームとサイネージ配信間にセッションが確立される。セッションが切断された場合は、サイネージ配信システムは速やかにサイネージプラットフォームへ再接続すること。

3.1.1. リクエスト

セッション確立のリクエスト形式を以下に示す。

本リクエストは、「図 1　システム全体の処理シーケンス図」における「セッション確立」に該当する。

- URL

```
ws(s)://<serverAddress>/notification?SignageServerId=<signageServerId>
```

- パラメーター

- ✓ SignageServerId
  サイネージ配信システムを一意に識別する文字列。
  GUID を指定する。

- ➢ ヘッダー
  Upgrade、Connection 等。
- ➢ ボディ
  なし。

リクエスト サンプル

```
GET http://example.com/notification?SignageServerId=6584c0f2-dca9-42cb-ad5b-e2ef5c2a043b HTTP/1.1
Host: example.com
Upgrade: Websocket
Connection: Upgrade
Sec-WebSocket-Key: k3CyFjeNOBNzFZx4blx9kg==
Origin: null
Sec-WebSocket-Version: 13
```

3.1.2. レスポンス

セッション確立のレスポンス形式を以下に示す。

本レスポンスは、「図 1　システム全体の処理シーケンス図」における「結果通知」に該当する。

- ➢ ステータス コード
  WebSocket への切り替えに成功した場合は 101（Switching Protocols）を返す。
  それ以外は接続結果に応じた既定の HTTP StatusCode を返す。
- ➢ ヘッダー
  Sec-WebSocket-Accept 等。
- ➢ ボディ
  なし。

レスポンス サンプル

```
HTTP/1.1 101 Switching Protocols
Upgrade: Websocket
Connection: Upgrade
Sec-WebSocket-Accept: WLjfOfV0ERDw6E/NUUaBpqXnHFc=
```

3.2. 更新通知（PUSH 型）

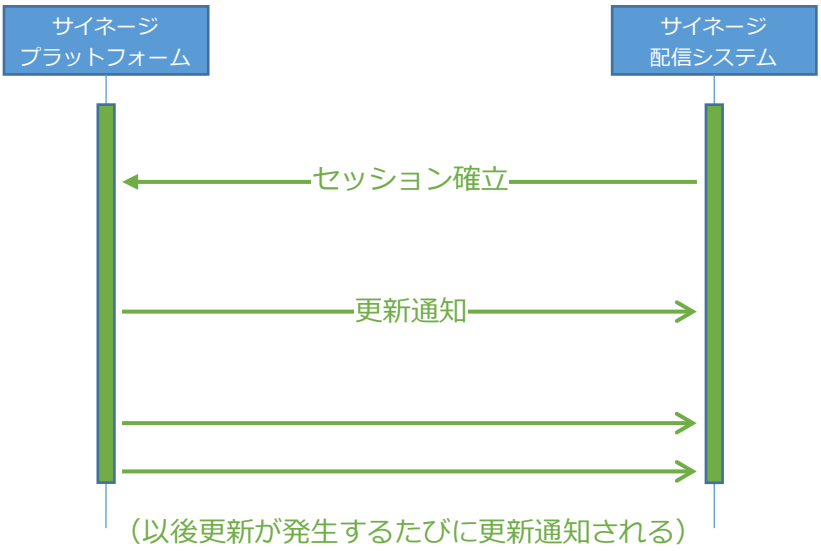

図 2　PUSH 型更新通知処理シーケンス図

PUSH 型で受ける更新通知は、「3.1 セッション確立」で確立したセッションの中で、サイネージプラットフォーム側から一斉配信のタイミングで送られる。

更新通知のメッセージ形式（WebSocket メッセージ）を以下に示す。本メッセージは、「図 1　システム全体の処理シーケンス図」における「更新通知」に該当する。

- 更新通知（WebSocket メッセージ）
  SPF-XML 形式のデータを送信する。

3.3. 更新通知（PULL 型）

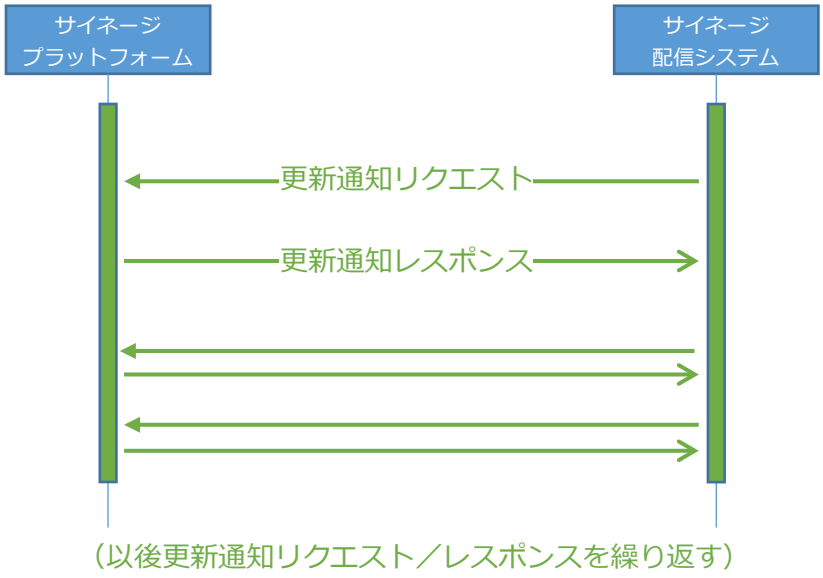

図 3　PULL 型更新通知処理シーケンス図

PULL 型で受ける更新通知は、POST リクエストでセッションを確立し、サイネージプラットフォーム側はそのセッションを維持、一斉配信が発生したタイミングでレスポンスを返す。なお WebSocket によるセッション確立は不要である。

更新通知のメッセージ形式（POST レスポンスメッセージ）を以下に示す。本メッセージは、「図 1　システム全体の処理シーケンス図」における「更新通知」に該当する。

- 更新通知（POST レスポンスメッセージ）
  SPF-XML 形式のデータを送信する。

3.4. コンテンツ取得要求

「3.2 更新通知」で取得した SPF-XML 内に記載されている URL へアクセスする。

3.4.1. リクエスト

コンテンツ取得要求のリクエスト形式を以下に示す。

本リクエストは、「図 1　システム全体の処理シーケンス図」における「コンテンツ取得要求」に該当する。

- URL

```
GET <「3.2 更新通知」で取得したコンテンツの URL>
```

- パラメーター

  なし。
- ヘッダー

  なし。
- ボディ

  なし。

3.4.2. レスポンス

コンテンツ取得要求のレスポンス形式を以下に示す。

- ステータス コード

  送信に成功した場合は 200（OK）を返す。

  それ以外は送信結果に応じた既定の HTTP StatusCode を返す。
- ヘッダー

  Content-Type 等。
- ボディ

  該当のコンテンツ。

3.5. コンテンツ取得結果送信

コンテンツの取得結果を送信する。

3.5.1. リクエスト

コンテンツ取得結果送信のリクエスト形式を以下に示す。

本リクエストは、「図 1　システム全体の処理シーケンス図」における「コンテンツ取得結果送信」に該当する。

- URL

```
POST http(s)://<serverAddress>/sendResult?SignageServerId=<signageServerId>
```

- パラメーター
  - ✓ SignageServerId
    サイネージ配信システムを一意に識別する文字列。
    GUID を指定する。
- ヘッダー

  Content-Length 等。
- ボディ

  以下の構造の XML。(application/xml)

フィールド	データ型	説明
distributionId	string	取得した SPF-XML の distributionId。
status	string	処理結果。 成功時は “succeeded”。 失敗時は “failed”。
failureReason	string	失敗理由。(処理成功時は省略。) 内容は自由記述。

リクエスト サンプル

```
POST http://example.com/sendResult?SignageServerId=6584c0f2-dca9-42cb-ad5b-e2ef5c2a043b HTTP/1.1
Host: example.com
Content-Length: 162
Origin: http:// example.com
Content-Type: application/json

<?xml version="1.0" encoding="utf-8"?>
<result>
  <distributionId>43eafbb5-94c3-45d6-bf2f-32c5c0d02110</distributionId>
```

```
  <status>succeeded</status>
</result>
```

3.5.2. レスポンス

コンテンツ取得結果送信のレスポンス形式を以下に示す。

- ステータス コード
  送信に成功した場合は 200（OK）を返す。
  それ以外は送信結果に応じた既定の HTTP StatusCode を返す。
- ヘッダー
  なし。
- ボディ
  なし。

レスポンス サンプル

```
HTTP/1.1 200 OK
```

4. SPF 表示依頼インタフェース仕様

SPF 表示依頼情報インタフェースとは、情報提供者向けのインタフェースであり、情報をサイネージプラットフォームへ入力するためのものである。情報提供者・サイネージプラットフォーム間の通信は、サイネージプラットフォームが情報提供者にリクエストする形を想定し、情報提供者が接続に必要なエンドポイントを用意する。

表 2　通信インタフェース一覧

名称	プロトコル	メソッド	URL
セッション確立	ws (wss)	–	~/connect
情報送信	ws (wss)	–	–
コンテンツ情報処理結果送信	http (https)	POST	~/sendResult

4.1. セッション確立

サイネージプラットフォームが、情報提供者に対しネゴシエーションを行うインタフェース。プロトコルは WebSocket を使用する。

当インタフェースでサイネージプラットフォームと情報提供者間にセッションが確立される。セッションが切断された場合は、サイネージ配信システムは速やかにサイネージプラットフォームへ再接続すること。

4.1.1. リクエスト

セッション確立のリクエスト形式を以下に示す。

本リクエストは、「図 1　システム全体の処理シーケンス図」における「セッション確立」に該当する。

- URL

```
ws(s)://<serverAddress>/connect?SignagePlatformId=<signagePlatformId>
```

- パラメーター

  SignagePlatformId

  サイネージプラットフォームを一意に識別する文字列。

  GUID を指定する。

- ヘッダー

  Upgrade、Connection 等。

- ボディ

  なし。

リクエスト サンプル

```
GET http://example.com/connect?SignagePlatformId=6584c0f2-dca9-42cb-ad5b-e2ef5c2a043b HTTP/1.1
Host: example.com
Upgrade: Websocket
Connection: Upgrade
Sec-WebSocket-Key: k3CyFjeNOBNzFZx4blx9kg==
Origin: null
Sec-WebSocket-Version: 13
```

4.1.2. レスポンス

セッション確立のレスポンス形式を以下に示す。

- ステータス コード
  WebSocket への切り替えに成功した場合は 101（Switching Protocols）を返す。
  それ以外は接続結果に応じた既定の HTTP StatusCode を返す。
- ヘッダー
  Sec-WebSocket-Accept 等。
- ボディ
  なし。

レスポンス サンプル

```
HTTP/1.1 101 Switching Protocols
Upgrade: Websocket
Connection: Upgrade
Sec-WebSocket-Accept: WLjfOfV0ERDw6E/NUUaBpqXnHFc=
```

4.2. 情報送信

情報送信は、「4.1 セッション確立」で確立したセッションの中で、情報提供者側からコンテンツ発行のタイミングで送られる。コンテンツ発行通知のメッセージ形式（WebSocket メッセージ）を以下に示す。

- 情報送信（WebSocket メッセージ）

  SPF-XML 形式のデータを送信する。

4.3. コンテンツ情報処理結果送信

コンテンツ情報の処理結果を送信する。

4.3.1. リクエスト

コンテンツ情報処理結果送信のリクエスト形式を以下に示す。

- URL

```
POST http(s)://<serverAddress>/sendResult?SignagePlatformId=<signagePlatformId>
```

- パラメーター

  SignagePlatformId

  サイネージプラットフォームを一意に識別する文字列。

  GUID を指定する。

- ヘッダー

  Content-Length 等。

- ボディ

  以下の構造の XML。(application/xml)

フィールド	データ型	説明
distributionId	string	取得した SPF-XML の distributionId。
status	string	処理結果。 成功時は “succeeded” 失敗時は “failed”
failureReason	string	失敗理由。(処理成功時は省略。) 内容は自由記述。

リクエスト サンプル

```
POST http://example.com/sendResult?SignagePlatformId=36cb6201-f500-4176-a89f-09c50c28adf2 HTTP/1.1
Host: example.com
Content-Length: 162
Origin: http://example.com
Content-Type: application/xml

<?xml version="1.0" encoding="utf-8"?>
<result>
  <distributionId>88658fe5-ff34-42a5-b439-3e106b861cdd</distributionId>
  <status>succeeded</status>
</result>
```

4.3.2. レスポンス

コンテンツ情報処理結果送信のレスポンス形式を以下に示す。

- ステータス コード
  送信に成功した場合は 200（OK）を返す。
  それ以外は送信結果に応じた既定の HTTP StatusCode を返す。
- ヘッダー
  なし。
- ボディ
  なし。

レスポンス サンプル

```
HTTP/1.1 200 OK
```

以上