

レグザクラウドサービス「TimeOn」のプラットフォーム技術

Platform Technologies Realizing "TimeOn" Regza Cloud Service

海邊 裕 深井 祐介 嵯峨 砂与子

■ KAIBE Hiroshi

■ FUKAI Yusuke

■ SAGA Sayoko

レグザクラウドサービス「TimeOn」(以下、TimeOnと略記)を実現するための新たなプラットフォームを開発した。これは、クラウドサーバを利用したネットワークサービス側のプラットフォームと、WebSocketサーバを組み込んだテレビ側のプラットフォームで構成している。これらは、最近広く使われるようになったHTML5 (Hypertext Markup Language 5) 並びにクラウドサーバ関連技術に基づくプラットフォーム技術を適用している。これにより、既存のサービスを連携させるといふ、TimeOnが目指す新たなサービスを生み出すことができる。

To realize the "TimeOn" Regza cloud service, Toshiba has developed two new platform technologies based on HTML5 (HTML: Hypertext Markup Language) and cloud technologies: a network service using cloud technology, and a WebSocket server implemented on TV products. These technologies make it possible to offer a base service platform in cooperation with existing multiple services.

1 まえがき

近年、ソーシャルネットワークサービスが広く世の中に浸透してきており、Web関連技術を基盤として発展を続けている。

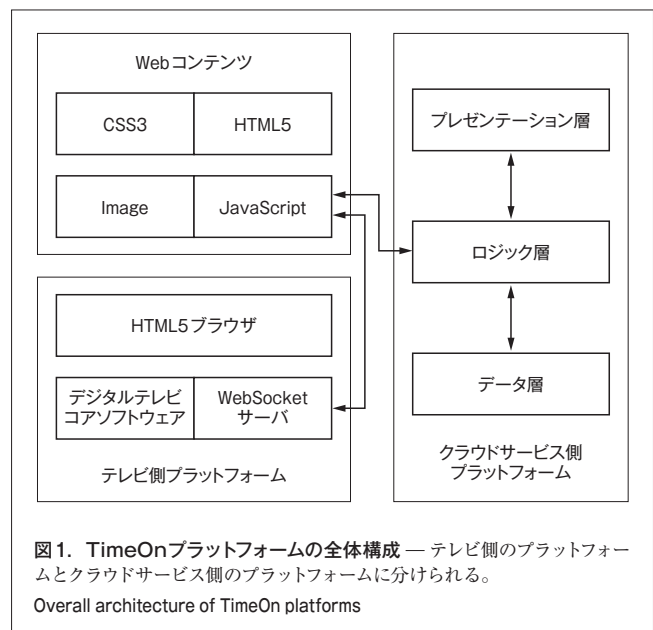
東芝は、今回、このサービスとの連携を目指すTimeOnを実現するために、HTML5をはじめとするWebに関する技術を導入して連携のプラットフォームを確立し、組み込み型の製品であるテレビに適用した。

ここでは、TimeOnの実現にあたって開発したWeb関連技術のうち、クラウドサービスのプラットフォーム並びにテレビに組み込まれたプラットフォームについて述べる。

2 全体構成

TimeOnのプラットフォームは、テレビ側のプラットフォームとクラウドサービス側のプラットフォームに分けられる。TimeOnのプラットフォームの構成について、その概要を図1に示す。

クラウドサービス側のプラットフォームは、静的なWebコンテンツを配布するプレゼンテーション層、動的なWebコンテンツを提供するロジック層、及びデータベースを利用し永続的なデータ保管を行うデータ層から構成される。静的なWebコンテンツとは、HTML5、CSS3 (Cascading Style Sheet 3)、JavaScript、並びにImageという形式のデータを意味する。一方、動的なWebコンテンツとは、JSON (JavaScript Object Notation) 又はXML (Extensible Markup Language) の形式でロジック層からWebブラウザに転送され、随時処理されるデータを意味する。



このようにクラウドサービス側のプラットフォームは、一般に知られている3階層モデルを採用することで、クラウドサーバの提供者を容易に選択できるように設計した。

一方、テレビ側のプラットフォームは、WebSocketサーバを採用することで、テレビ上のGUI (Graphical User Interface) からも、また、タブレットなどの外部機器からも、同じようにテレビの機能を利用することができるようになった。

次章以降では、それぞれのプラットフォームについて詳細を述べる。

3 クラウドサービス側プラットフォーム

3.1 概要

クラウドサービス側のプラットフォームは、クラウドサーバの供給者から提供されるプラットフォームの要素を組み合わせることで構築されている。これらの要素を使った分だけ費用を支払う契約を選択することで、可用性や応答性を目標に合わせながら、最小の費用でサービスを運用できる。このことが、クラウドサーバを用いて得られる最大の恩恵である。ただ、供給者から提供される構成要素は多岐にわたるため、それらの組合せ方に技術が必要となる。

3.2 プラットフォームの要素

ここでは、TimeOnのクラウドサービス側プラットフォームで用いたそれぞれの要素について述べる。

3.2.1 ホスティングエリア クラウドサーバの供給者が管理するネットワーク及びハードウェア設備であり、世界の各地にデータセンターとして存在する。TimeOnでは、国内のホスティングエリアを利用した。

3.2.2 仮想ネットワーク ホスティングエリアに存在するネットワーク設備を設定することで、物理的な配線とは異なる仮想ネットワークが提供される。TimeOnでは、セキュリティ性を高めるため、利用する仮想サーバ間だけに限定された仮想ネットワークを利用している。

3.2.3 仮想サーバ ホスティングエリアに存在するハードウェア設備上で実行される仮想コンピュータである。仮想コンピュータになることで、障害発生時の移送や負荷増大時の複製化が容易になる。TimeOnでは、様々な容量の仮想サーバを同時に用いることでサービスを構築している。

3.2.4 ロードバランサ ネットワークの負荷に応じて仮想サーバを複製化することが必要になるが、この負荷の振分けを担当する機構である。また、ネットワークサービスはHTTPS (Hypertext Transfer Protocol Security) によるセキュリティ性の向上が求められるため、ロードバランサでHTTPSを処理している。TimeOnでは、プレゼンテーション層とロジック層のサービスそれぞれでロードバランサを利用している。

3.2.5 データベース 永続性が必要とされるデータを格納するためのリレーショナル データベースである。TimeOnでは、障害耐性を確保するため、待機冗長構成のデータベースを利用している。

3.2.6 仮想ストレージ ファイル保存のための可用性が高いデータ記録サービスである。データベースと異なり、データ間の演算は提供されないが容量に対しての運用コストが低い。TimeOnでは、画像データの保管やログの保管に仮想ストレージを利用している。

3.3 機能の構成

ここでは、TimeOnのクラウドサービス側プラットフォームのサービス機能について述べる。

3.3.1 プレゼンテーションサービス Webの静的なコンテンツを提供するサーバである。すなわちプレゼンテーション層を担当するサービスで、ロードバランサに接続された仮想サーバにWebサーバを配備し実現している。

3.3.2 アカウント管理サービス テレビやタブレットそれぞれの固有識別子と登録済みのユーザーアカウントを管理するサービスである。また、これらの識別子やユーザーアカウントに付随する情報、更にこれらの情報間の関連も管理する。3階層モデルとして、ロジック層とデータ層を提供する。具体的には、ロードバランサに接続された仮想サーバにWebアプリケーションサーバを配備している。更に、データ層の実現のために、データベースと仮想ストレージを利用している。

3.3.3 番組データなどの管理サービス 録画予約の情報やテレビ番組の情報を管理し、かつ、それぞれの関係も管理するサービスである。こちらも、ロジック層とデータ層を提供するため、前記のアカウント管理と同様の要素から構成される。

3.3.4 運用監視サービス 監視サービスを導入することで、TimeOnの構成要素を常時監視している。なお、万一の障害発生時には速やかに復旧できるように体制作りも重要である。

3.4 運用状況

TimeOnのサービスは、2012年10月末から一般向けに運用が開始されており、2013年1月時点での登録デバイス数は10,000台を超過し、サービス規模が順次拡大している状況である。現時点では、目標としている運用コストと応答性能のバランスは取れているが、今後の負荷の変動に合わせて更にバランスが向上するよう、引き続きプラットフォームの改善を進めていく。

4 テレビ側プラットフォーム

4.1 概要

HTML5の普及に伴い、Webブラウザ上で高機能なWebアプリケーションが開発できるようになっている。これまで当社は、デジタルテレビなどに組み込まれた機能と協調するWebアプリケーションを動作させるため、WebKitの映像製品への適用⁽¹⁾や、Webブラウザを用いたGUIの試作⁽²⁾などに取り組んできた。これらを基に、今回、TimeOnサービスをデジタルテレビ上で動作させるテレビ側プラットフォームを構築した。

テレビ側プラットフォームは、WebSocketを用いてテレビの機能を公開する仕組みにした。Webブラウザ上で動作する

JavaScriptからテレビ内部に搭載したWebSocketサーバへ接続し、JSONを基にした当社独自のプロトコルを用いてテレビ機能への要求やテレビ機能からの通知を受け取る。テレビ側プラットフォームの構成を図2に示す。

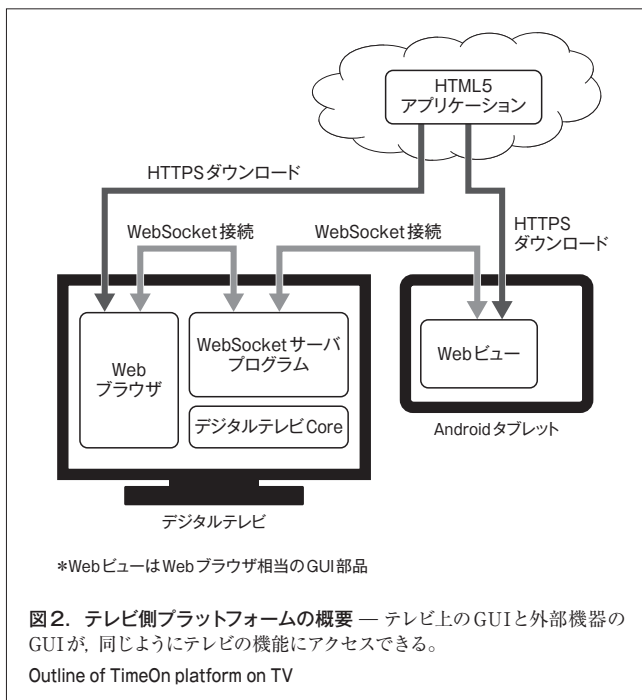
4.2 WebSocketを用いる方式の利点

Webブラウザへテレビ機能を公開する方式には、Webブラウザを拡張する方式やHTTPを用いる方式などがある。それらと比較しながら、WebSocketを用いる方式の利点を以下に挙げる。

Webブラウザを拡張する方式と比較した場合、外部機器との連携が可能という利点がある。これにより、AndroidTM(注1)を搭載したタブレット上で動作する、RZクラウドのようなコンパニオンアプリケーションからテレビ機能にアクセスできるようになっている。

WebSocketと同じくテレビ内部へサーバを搭載する方式の中で、HTTPを用いる方式と比較した場合、利点の一つは高速という点である。毎回コネクションを確立する必要があるHTTPと比較し、WebSocketはコネクションを確立したまま通信を継続できるため、オーバーヘッドが少なく高速である。もう一つの利点は、双方向通信ができるという点である。テレビ機能には、チャンネル変更通知など、アプリケーションが取得したいイベントがいくつかある。HTTPでもロングポーリングなどを行えば可能ではあるが、短時間に連続してイベントが起こった場合などに、イベントが欠落するおそれがある。

これらの理由から、WebSocketを用いた方式を選定した。



(注1) Androidは、Google Inc.の商標。

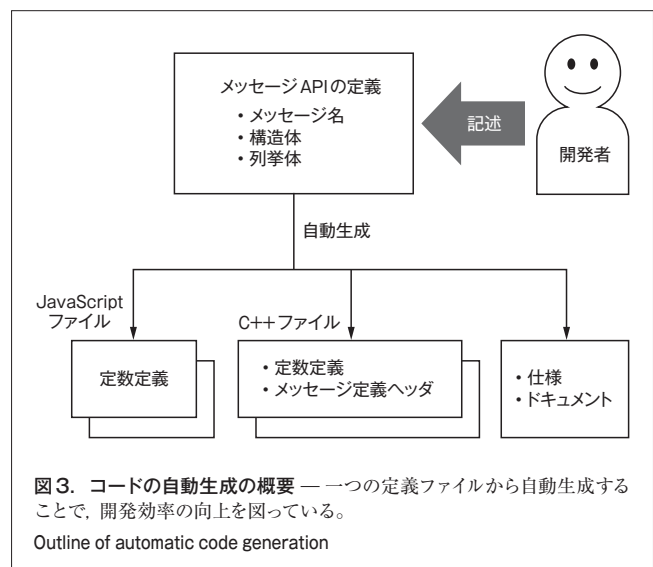
4.3 製品化における工夫

ここでは、製品化にあたり工夫した点について述べる。

4.3.1 共通メッセージAPIの定義 仕向地別や製品別に様々な要求仕様があるが、それらを包括して同じメッセージAPI (Application Programming Interface) で取り扱えるよう600個を超える共通的なメッセージAPIを定義した。各製品はこの中から必要なサブセットを選択して実装する。今回のテレビ側プラットフォームでは、約80個のメッセージAPIを実装している。このように共通メッセージAPIを定義したことで、異なる製品に対しても、録画機能など共通の機能に対して同様にアクセスできるため、アプリケーションの大部分を共有化することが可能になる。

4.3.2 独自プロトコルの開発 多くのメッセージAPIを実装しているため、JSONにメッセージ名などを含めると、通信時間の増加や文字列比較のための処理時間の増加など効率の低下を招く。したがって、独自のプロトコルを定め、テレビ側にメッセージ名など定数定義したファイルを用意し、アプリケーション側で用いるJavaScriptライブラリから通信開始時にそのファイルを取得して用いることで、通信時間や処理時間の高速化を図った。また、この仕組みにより、新製品でメッセージAPIを追加した場合でも、定数定義ファイルにメッセージ名が存在するか確認することでサポート状況がわかるようになっている。

4.3.3 コードの自動生成による開発効率向上 各メッセージAPIについて、定数定義ファイルやデジタルテレビ側で公開する機能のためのコードなどを、同期して誤りなく記述する必要がある。そのため、メッセージAPIの定義を記述したファイルから定数定義ファイルやテレビ側の公開機能実装用メッセージ定義ヘッダなどを自動生成することで、開発効率の向上を図った。コードの自動生成の概要を図3に示す。



4.3.4 セキュリティの確保 テレビ側プラットフォーム

を取り入れることで、テレビ外部からテレビ機能にアクセスすることが可能になる。そこで、悪意のある攻撃から守るため、正当な手続きを踏んで開発したアプリケーションだけにアクセスを許す仕組みを搭載した。

メッセージAPIにはシステムの初期化など危険性の高いものから、チャンネル情報の取得など危険性の比較的低いものまである。そのため、各メッセージAPIをカテゴリー分けして危険度に合わせたレベルを設定し、アプリケーションにはカテゴリーごとにアクセス可能なメッセージAPIレベルを設定し、使用できるメッセージAPIを制限する仕組みを設けた。

また、外部機器からテレビ機能にアクセスする際には、テレビと外部機器の間で認証処理を実行することで、外部機器からのアクセスの正当性を確認している。

5 あとがき

TimeOnを実現するために開発した、クラウドサービス側のプラットフォーム技術とテレビ側のプラットフォーム技術について述べた。

それぞれは汎用的な標準技術に基づいており、将来の製品ラインアップの拡充及び他製品への展開が容易になると期待される。また、これらのプラットフォーム技術が世の中のネットワーク技術の動向に常に連動するよう、更なる改良に努めていく。

文 献

- (1) 平野 裕 他. Webブラウザエンジン WebKitの映像製品への適用. 東芝レビュー. 67, 8, 2012, p.19-22.
- (2) Fukai, Y. et al. "Web Browser based GUI for TV". The 1st IEEE Global Conference on Consumer Electronics (GCCE 2012). Tokyo, 2012-10, IEEE. p.579-580.



海邊 裕 KAIBE Hiroshi

デジタルプロダクツ&サービス社 プラットフォーム&ソリューション開発センター プラットフォーム・ソリューション開発第五部参事。クラウドプラットフォームの開発に従事。
Platform & Solution Development Center



深井 祐介 FUKAI Yusuke

デジタルプロダクツ&サービス社 プラットフォーム&ソリューション開発センター プラットフォーム・ソリューション開発第三部主務。組込みGUIプラットフォームの開発に従事。
Platform & Solution Development Center



嵯峨 砂与子 SAGA Sayoko

デジタルプロダクツ&サービス社 プラットフォーム&ソリューション開発センター プラットフォーム・ソリューション開発第三部主務。組込みGUIプラットフォームの開発に従事。
Platform & Solution Development Center