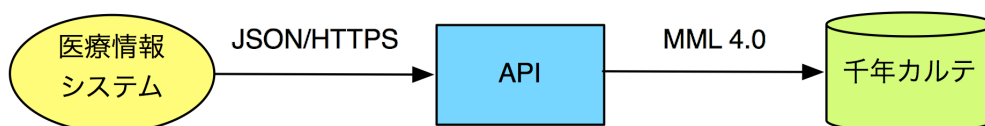


1. MML4アップコンバータとは

- 電子カルテなどの医療情報システムに簡単なAPIを提供し、千年カルテが必要とするMML4.x形式のデータを生成するAPIサーバー。
- APIのスタイルとしてはREST、データ形式としてはJSONを使用。



2. 目的

- 医療情報システムベンダーの千年カルテ対応の開発コストを下げる。

プロジェクト側のメリット

- データ収集の効率化を図る。
- MMLインスタンスの品質のバラツキを是正する。
- デフォルトのアクセス権設定
- プロジェクトに必要なデータをAPIで規定（名寄せのためのヨミ、全角にする、区切り文字等）

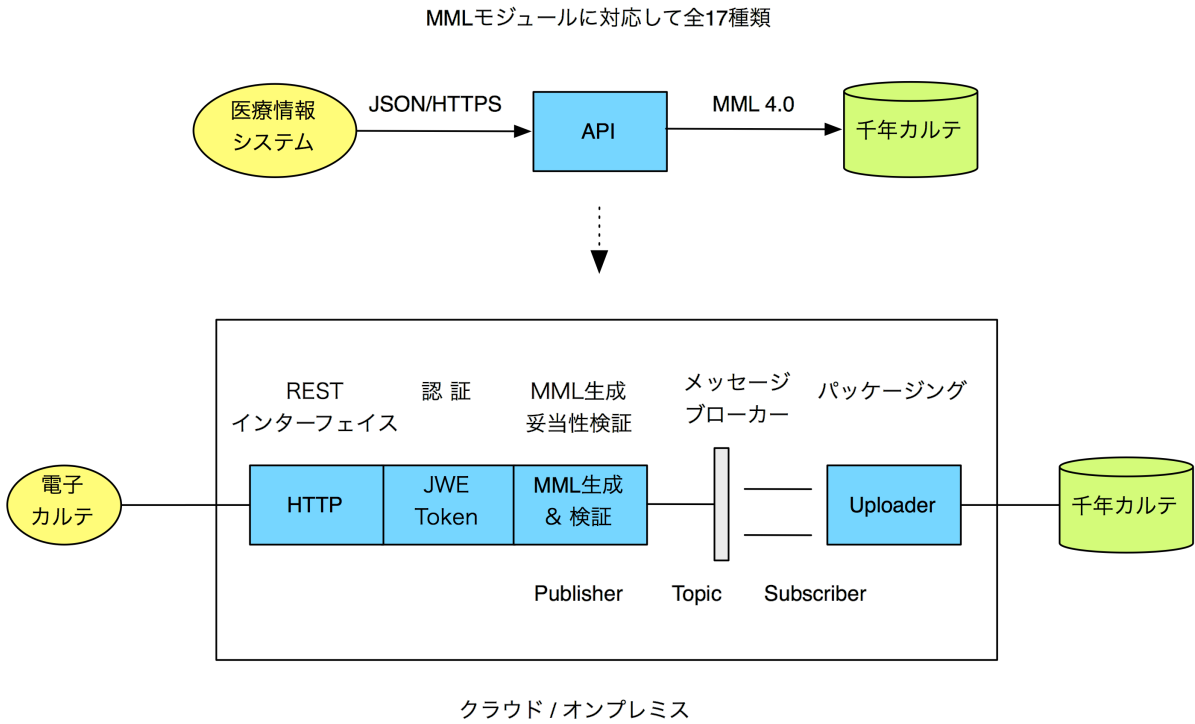
ベンダー側のメリット

- 千年カルテプロジェクトに固有なデータ設定が隠蔽されているので医療情報自体に専念できる
- MML/openEHRの理解に役立つ

3. 特長

- 医療情報システムの種類（業務の種類やプログラムの実装言語）に依存しない。
- 病院の規模を問わず、大病院からクリニックまでカバー できる。
- 安全に運用できる。

4. APIサーバーの構成要素



プログラムモジュール	機能
HTTP RESTインターフェイス	HTTPによるRESTエンドポイント提供
アプリケーションの認証	APIを使用するクライアントプログラムを認証する (認証されたアプリケーションのみがAPIを使用できる)
MML4.x生成	POSTされたAPIデータからMML4.xを生成
MML 妥当性検証	生成されたMMLのバリデーションチェック
Message Broker	短時間で大量のメッセージを処理するための機能
アップローダへのパッケージング	千年カルテへの送信要件を満たすようにパッケージを作成し、アップローファへ引き渡し

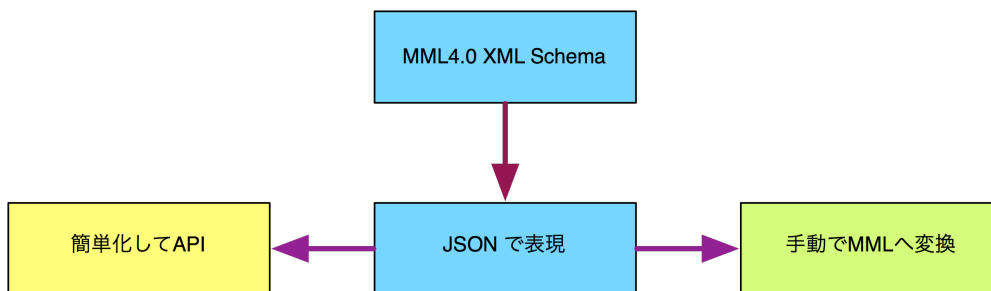
5. API の種類

- MMLのモジュール毎にAPI を定義 (全部で17種類)

MML モジュール名称	API の データ形式 JSON	REST エンドポイント
患者情報	simplePatient	/mml/api/v1/patientInfo
健康保険情報	simpleHealthInsurance	/mml/api/v1/healthInsurance
診断履歴	simpleRegisteredDiagnosis	/mml/api/v1/registeredDiagnosis
生活習慣	simpleLifestyle	/mml/api/v1/lifestyle
基礎的診療情報	simpleBaseClinic	/mml/api/v1/baseClinic
初診時特有情報	simpleFirstClinic	/mml/api/v1/firstClinic
経過記録	simpleProgressCourse	/mml/api/v1/progressCourse
手術記録	simpleSurgery	/mml/api/v1/surgery
臨床サマリー	simpleSummary	/mml/api/v1/summary
検歴情報	simpleTest	/mml/api/v1/test
報告書	simpleReport	/mml/api/v1/report
紹介状	simpleReferral	/mml/api/v1/referral
バイタルサイン	simpleVitalSign	/mml/api/v1/vitalsign
体温表	simpleFlowSheet	/mml/api/v1/flowsheet
処方箋	simplePrescription	/mml/api/v1/prescription
注射記録	simpleInjection	/mml/api/v1/injection
透析記録	simpleHemodialysis	/mml/api/v1/hemodialysis

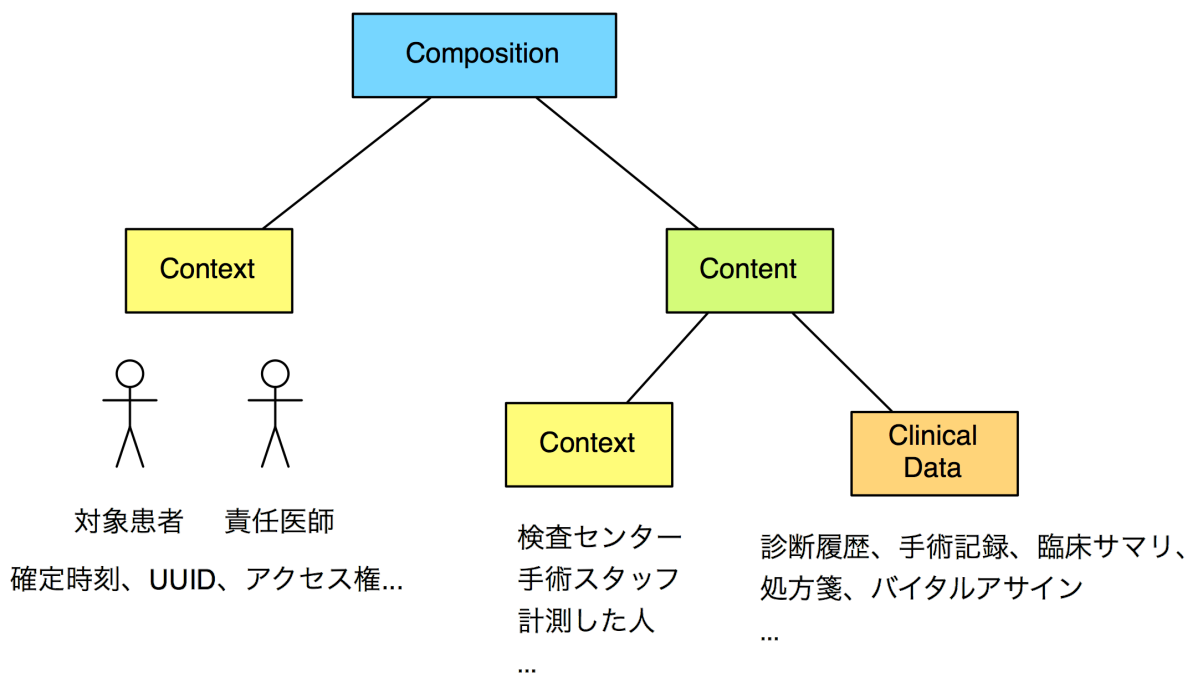
6. データ形式 JSON の作成手順

- MML4.0の XML Schema を JSON で表現 => J2
- J2 を手動でMMLに変換
- J2 を簡単化しAPIとして提供



7. コンポジションの導入

- 情報をコンテキスト（文脈）とコンテンツ（臨床データ）に分ける => MML/openEHR のトップ構造
- API を通過するデータはコンポジションにラッピングする



MM4 アップコンバータ

コンテキストになる情報（医学イベントが起こった時の文脈 ～ MML の docInfo に相当）

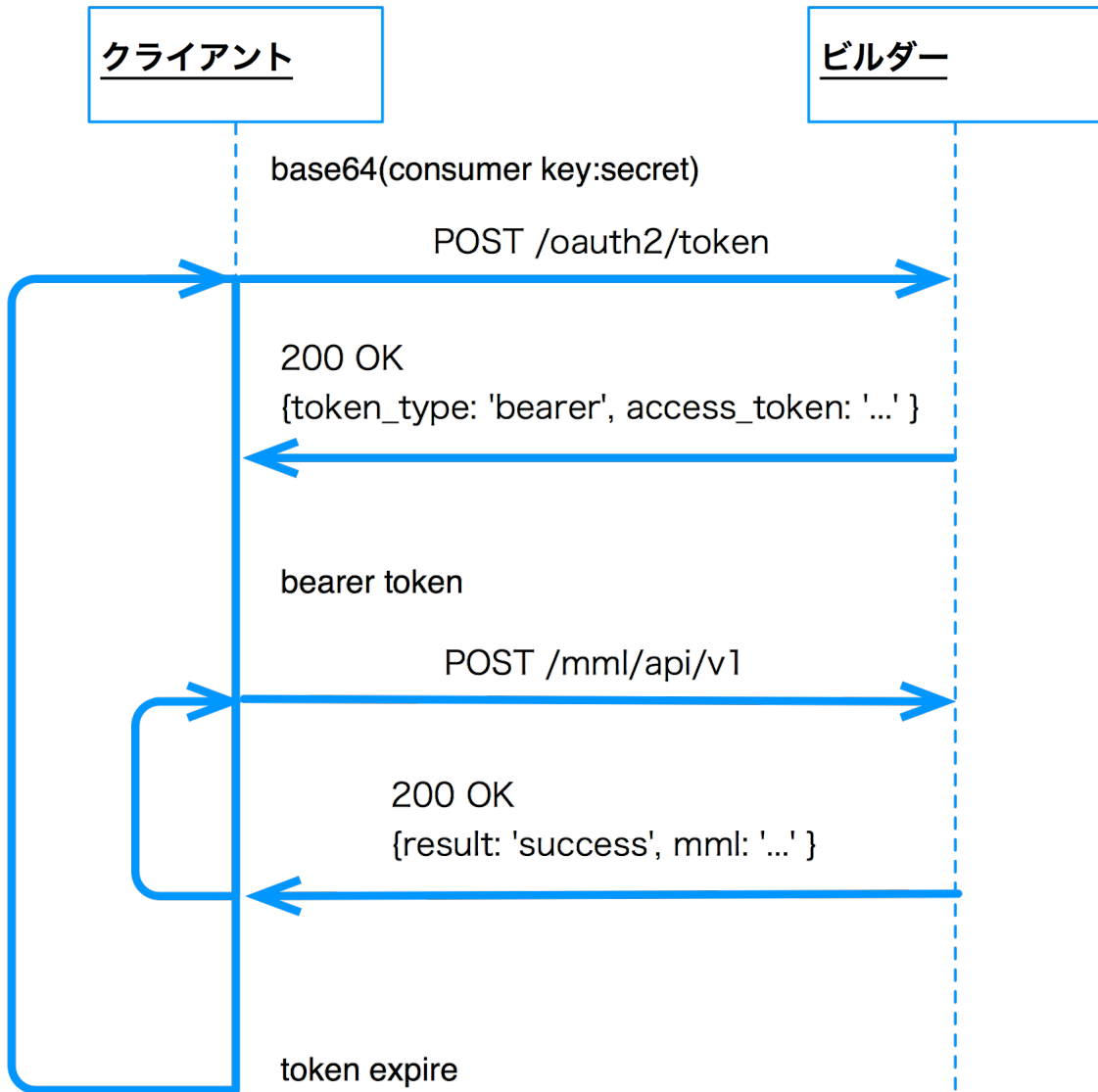
- 対象患者、責任医師、カルテの確定日、文書のUUID、アクセス権 ...

コンテンツになるデータ（～ 臨床データ）

- 診断履歴、手術記録、臨床サマリ、検査結果、処方箋、バイタルサイン...
- コンテンツはコンテキストも含む（手術スタッフ、検査会社、バイタルサイン測定者…）

8. 安全性-アプリケーションの認証

- APIは認証されたアプリケーションしか使用できない機能を組み込む
- プロジェクトから consumer key と secret key を提供
- アプリケーションは consumer key と secret key でHTTP BASIC認証を受ける
- 認証に成功すると Access Token が発行される
- アプリケーションは Access Token をHTTP Headerにセットして APIコールを行う
- システムは有効なAccess TokenであればAPIの実行を許可する
- Access Token には有効期限があり、失効したら認証からやり直す



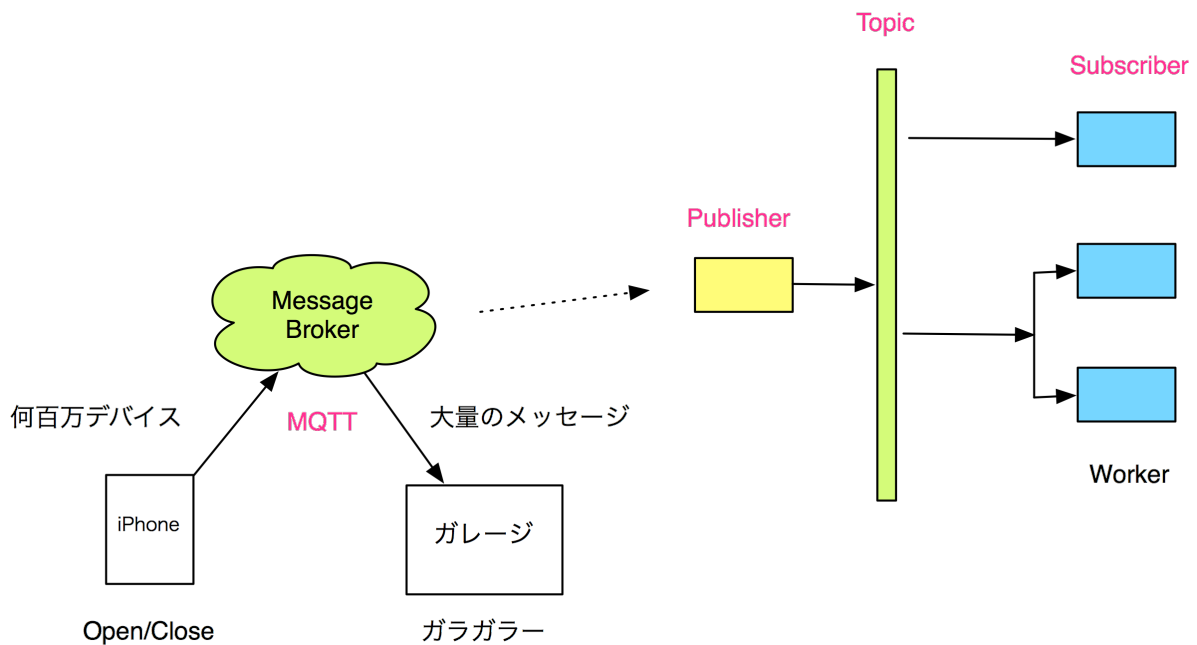
9. 認証のための標準

機能	仕様	備考
認証、アクセス手順	RFC6749	Client Credentials Grant flow of the OAuth 2 specification
Tokenの種類	RFC7516	JWE (JSON Web Encryption) 暗号化 (AES/GCM/256) モード (direct)
Token が暗号化する payload	-	発行日時、有効期間、アプリケーション情報 (ベンダー名、ア

		アプリケーション名、バージョン、 連絡先...)
--	--	-----------------------------

10. 病院の規模を問わない実装

- オプトアウト方式 => 全患者のデータが発生
- 作成タイミング: 一日の診療終了時 => 短時間に大量のデータが発生
- アップローダにはファイル渡し => 時間がかかる
- Message Broker を採用



11. 開発言語等

- Node.js Ver.6.10 以降
- HTTP Server Express 4.14 以降

12. 2016年度まとめ

成果

- 紹介サイトの開設 <https://1000-builder.au-syd.mybluemix.net/>
- API ガイド Wiki 提供 https://github.com/mbot-dev/1000_builder/wiki

MM4 アップコンバータ

- ソースコード公開 https://github.com/mbot-dev/1000_builder
- 2社に開発サーバー提供
- 本稼働サーバー設置
- 千年カルテプロジェクトに固有な運用規則の隠蔽（医療機関、診療科コード、文書ID、アップロード時のファイル命名規則、ZIP化方法等々）
- HIS に保存されている JSONデータの取り込み（設計:糸先生）

問題点

- HIS側の事情で必須要素が空で出力 => API 通過、MMLバリデータも合格 => プロジェクト側が受け取れない
- 例) 基礎的診療情報の血液型 ABO 型は必須、HISの都合でその他にセット
`<abo></abo>`
`<other>ここに値</other>`
- バリデータとしては正常
- API でブロックするか千年側でこの場合でも受け取る必要がある
- 上記のような問題が発生し、結果的にプロジェクト側のチェックが必要 => 負荷軽減に繋がらなかった

今後

このサーバーをクラウドに配備し、医療情報の収集サービスとして提供することを提案する。その際は、検査センター、介護施設、自宅の IoT 機器、他の医療情報交換規格からのデータも扱えるようにし、様々なソースとシンク（千年カルテ）を結ぶコネクタとしてさらなる抽象化を図りたい。そのための基盤となる技術、メッセージブローカー、スケールアウト、サービスを利用するベンダー側のアプリケーションを認証する方法についても、今回の開発で見通しを得た。

