



於 Seagaia

ORCA: Online Receipt Computer Advantage

ORCA（日医標準レセプトソフト）の連携機能

2017年5月19日
日本医師会ORCA管理機構（株）
営業企画部長 福田知弘



ORCAプロジェクト

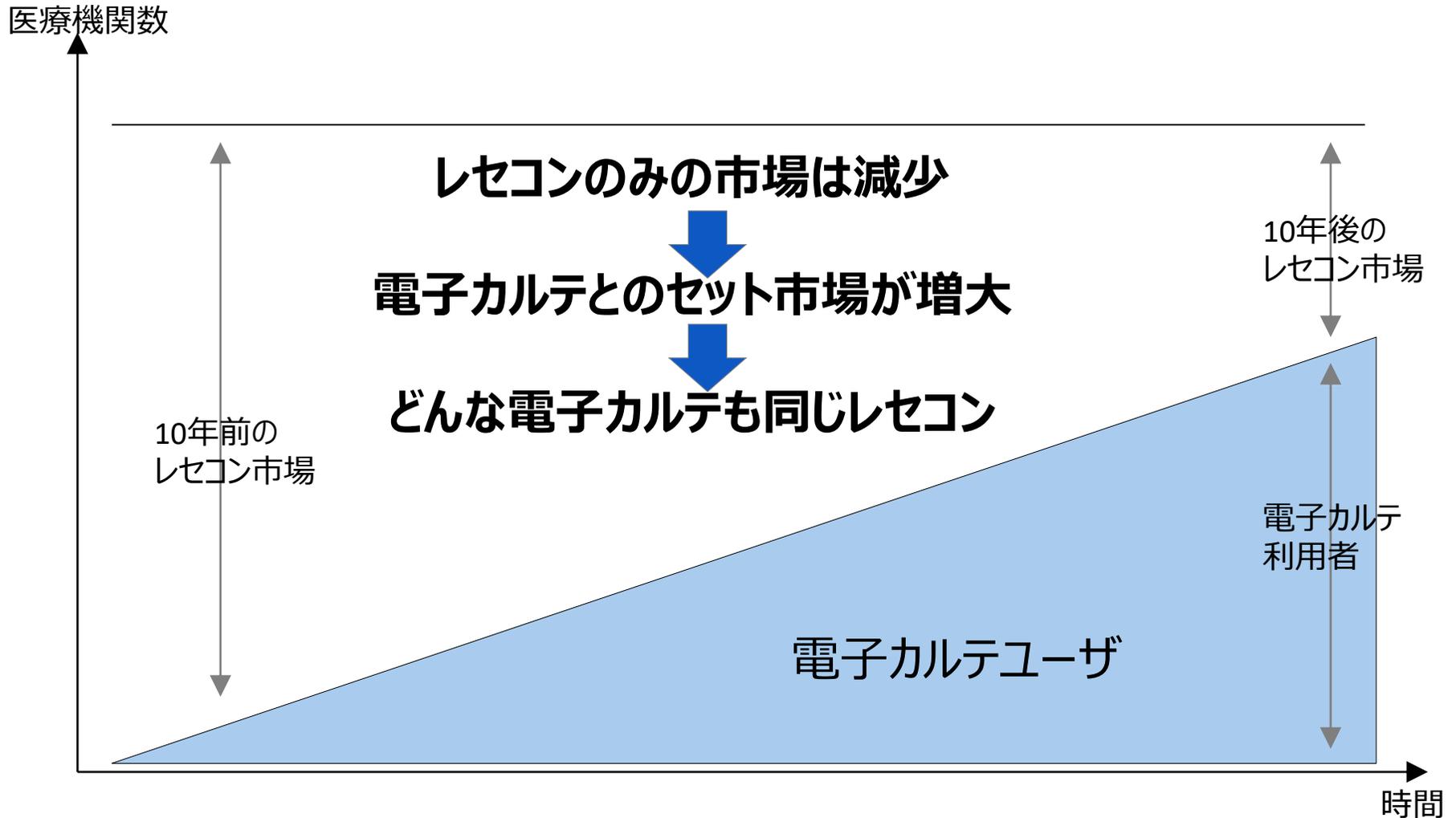
【日医 I T 化宣言2016】

- 安全なネットワークを構築するとともに、個人のプライバシーを守ります
- 医療の質の向上と安全の確保を I T で支えます
- 国民皆保険を I T で支えます
- 地域医療連携・他職種連携を I T で支えます
- 電子化された医療情報を電子認証技術で守ります。





レセコン市場の変化





会計処理エンジンのイメージ

利用者がサーバを意識しなくても利用できる環境



会計処理機能をクラウドサービス化し必要時に利用





ORCAと電子カルテの現状

- 調査期間：2016年11月18日～22日
- 対象：稼働中の電子カルテの数を各メーカーに問い合わせ

結果（40社中36社の集計結果）

総数	3,937件	(ORCAユーザ	15,978件	:	導入率	24,6%)
無床診療所	3,645件	(ORCAユーザ	13,631件	:	導入率	26,7%)
有床診療所	213件	(ORCAユーザ	1,734件	:	導入率	12,3%)
病院	79件	(ORCAユーザ	613件	:	導入率	12,9%)

ORCAとのインタフェース（36製品を対象）

CLAIMのみ	9製品
APIのみ	2製品
ODBCのみ	2製品
CLAIM + API	6製品
CLAIM + ODBC	7製品
API + ODBC	1製品
CLAIM+API+ODBC	4製品
独自 (glclient改)	5製品 (CLAIMやAPI、ODBCと複合で利用)

回答なしの電子カルテメーカーのユーザ数を含めると、ORCAユーザにおける無床診療所の電子カルテ導入率は概ね30%と推定される
(市場調査と同じレベル)

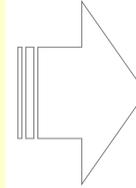
接続方法は想像以上にバリエティに富んでいた



ORCAの連携機能の標準化と強化

【既存のインタフェースの弊害】

- CLAIM … 規格の停滞、疎結合
- ODBC … DBの構造変更の影響
- DBS … ORCA独自仕様
- その他 … 各社独自開発

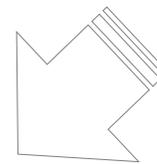


【次期インタフェース】

- 標準的、汎用的
- DBの構造変更柔軟に対応
- 密結合、イベント通知
- クラウド対応

【主要キーワード】

- HTTP、HTTPS
- XML、JSON
- Websocket
- Push通知



A P I
(Application Interface)



API協議会の設立とワーキング

APIの規格検討と普及活動

電子カルテを含めたORCAと連携するアプリケーションに必要なAPI（HAORI）の整備

- 新規API（HAORI）で真に必要な機能の選別
- API（HAORI）の利用方法（手続き）等の共有
- API（HAORI）を世に出す前に完成度を高める



連携アプリ毎にAPI（HAORI）に求める機能の整理

- 連携アプリで真に必要なAPI（HAORI）の選別
- 連携アプリを開発していくなかでのAPI（HAORI）の不明点、注意点の情報集約
- それらの情報を仕様書および解説書としてドキュメントに整理
- API（HAORI）の完成度と高めるための検証作業



運営組織としてAPI協議会を設立し、実業務は個別ワーキングとして活動していく



API協議会：今後のスケジュール



△：API協議会 定例会議



(~2016年6月)

項目 (画面相当)	API	HAORI
受付	2	
患者登録	2	
照会	5	
予約	3	
診療行為	5	
病名	1	
収納	1	
入退院登録	2	
入院会計照会	4	
退院時仮計算	1	
入院患者紹介	1	
マスタ登録	5	
システム管理情報	2	

2016年度開発



外来関連

2017年度開発



入院関連

(~2017年3月)

項目 (画面相当)	API	PUSH	HAORI
受付		1	
患者登録		1	4
照会	2	1	
診療行為	2	1	4
病名			2
収納			2
会計照会	3		2
明細書	3	2	



バッチ処理

(~2018年3月)

項目 (画面相当)	API	PUSH	HAORI
データチェック			
請求管理			
総括・公費			
日次統計			
月次統計			
システム管理情報			



【定義】

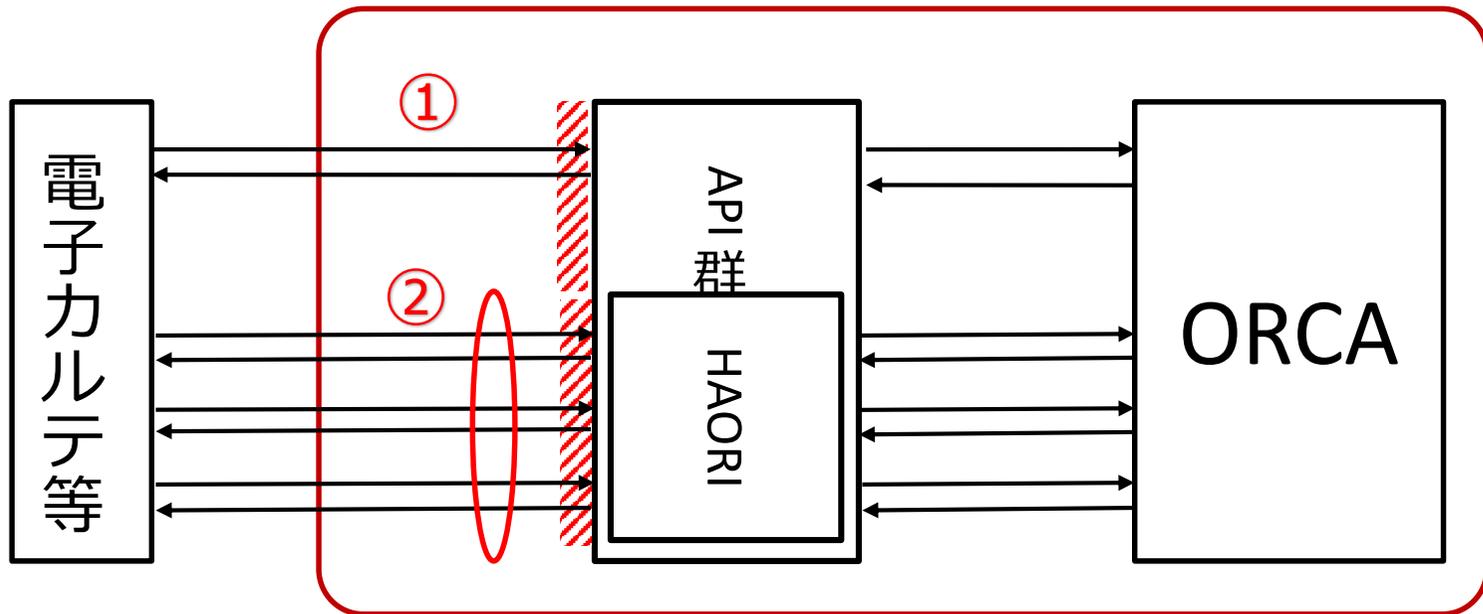
API

… 低度な連携機能（低額）

拡張API（HAORI）

… 高度な連携機能（高額）

【機能】

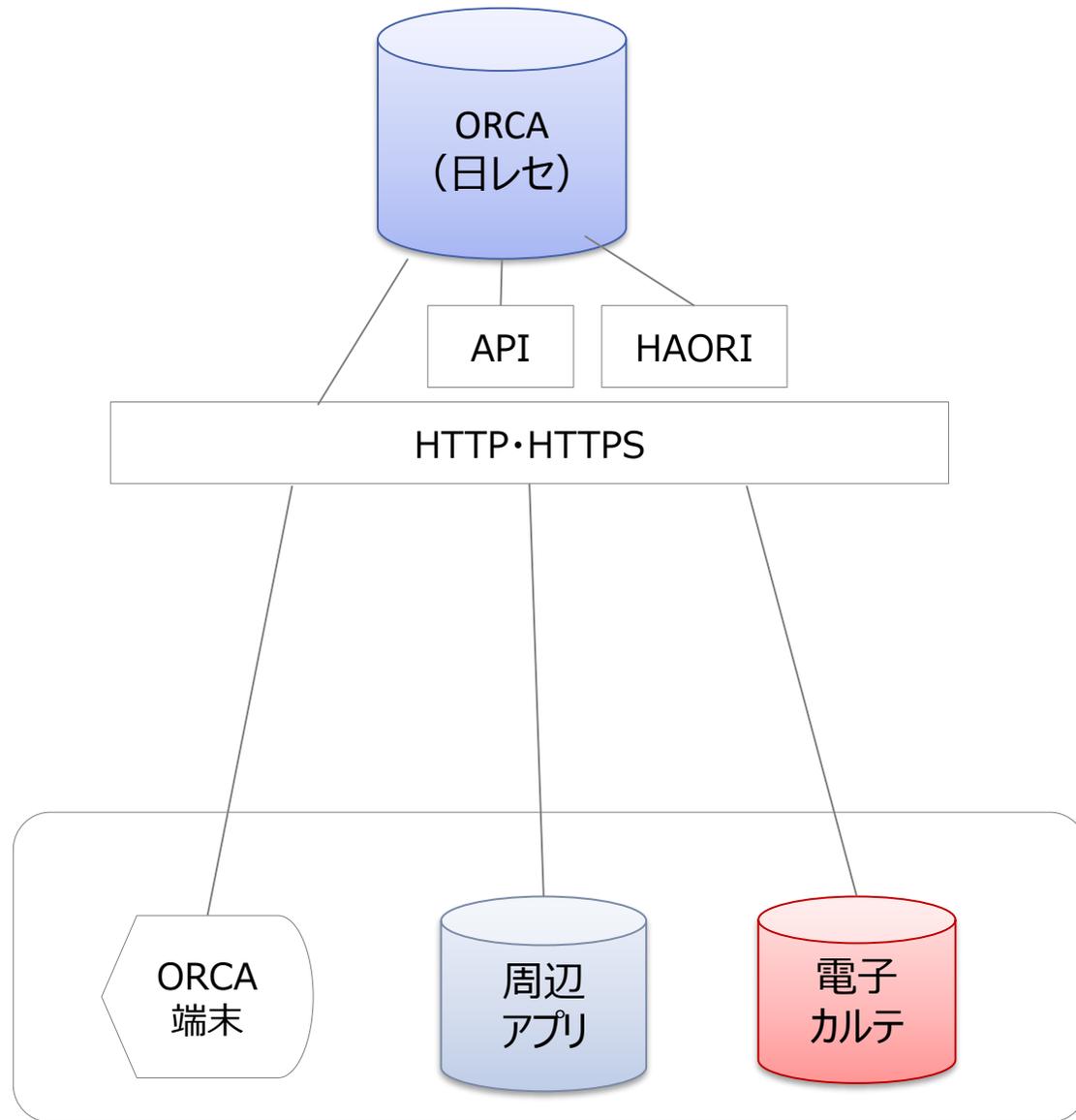


① APIは1回の処理で完結できるものをいう。

② HAORI（拡張API）は、一貫性をもった複数の処理を連続的に行うためのものをいう。
連携アプリケーション（電子カルテ等）はHAORIからのレスポンスで応じて次の処理を判断し、対話形式で一連の処理を行う。



ORCAのAPI連携





APIを使ったプログラミング

1. PUSH-APIによる連携・・・プログラミング

- PUSH-APIを使ったプログラムサンプルの提供
- CLAIM通信の仲介にPUSH-Exchangerを開発
- PUSH-Exchagnegrを理解することで、PUSH-APIの使い方を習得

2. PUSH-APIによる帳票印刷

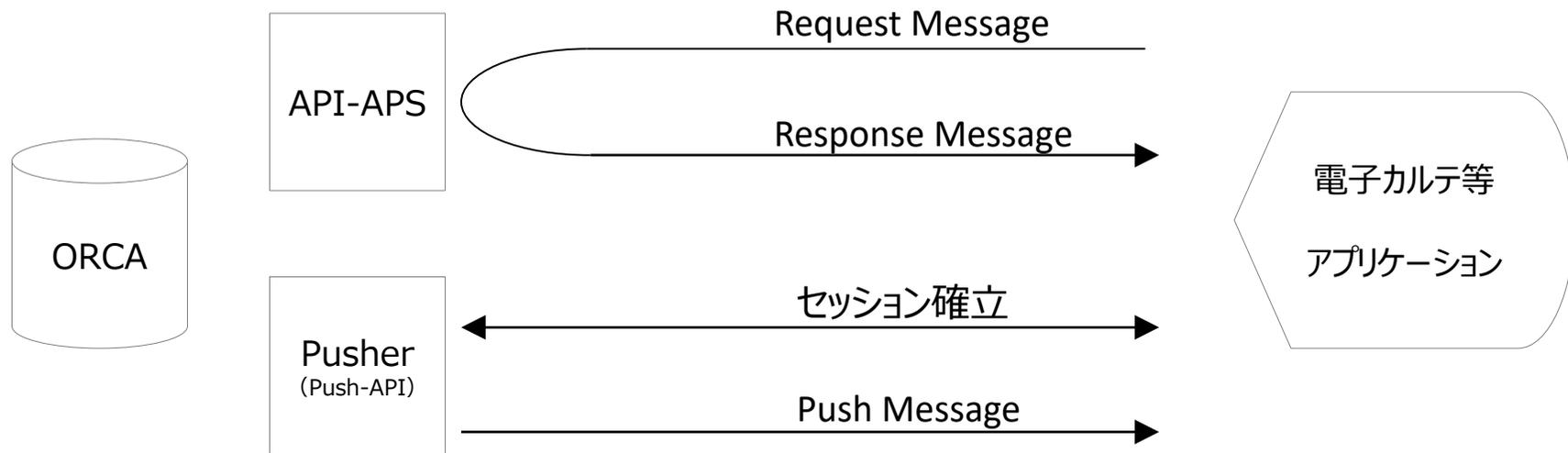
- PUSH-APIを使ってオンライン帳票を作成するサンプルプログラムの提供

3. WindowsでMonpeの印刷をする方法・・・プログラミング

- Linuxで作成したカスタマイズ帳票をWindowsで実現する仕組みの提供
- オンラインカスタマイズ帳票はクライアントで印刷イメージを作る
- データの取得はAPIで行い、必要な情報を印刷イメージに差し込み印刷をする



API (含むHAORI)による連携



- (1) APIは、電子カルテ等のアプリケーションからRequestを送信して、Responseを受信する手順である
- (2) Pusherは、電子カルテ等のアプリケーションがセッションを確立した後、ORCA側で通知するイベントが発生したときに、メッセージが送信される。ただし、通知されるメッセージには発生したイベントのキーとなる情報のみが格納されているので、通知メッセージを受信した場合は、該当する情報をAPIで取得しなければならない。



APIの使い方

```
def api_patientgetv2(conf,pid,ic_num,date)
  st,body = http_get(conf,"/api01rv2/patientgetv2?id=#{pid}") # 患者情報取得APIの呼び出し
  if st == "200" # HTTP_OK
    doc = REXML::Document.new(body)
    if doc.elements["xmlio2/patientinfores/Api_Result"].text.to_i != 0
      raise filter(doc.elements["xmlio2/patientinfores/Api_Result_Message"])
    end
    ptinf = parse_ptinf(doc) # XMLを解析して患者情報を取り出す
    ic = parse_insurance(conf,doc,ic_num,date) # 保険番号から保険者情報APIを呼び出して保険者情報
    return ptinf,ic
  else
    puts "patientgetv2 failer: #{st}"
    exit 1
  end
end
```



APIの使い方

レスポンスメッセージ (XML)を解析して帳票作成

診療報酬明細書(医科入院外)2国		平成28年 4月分	県番13	医コ12-3456-7	1医科	1国	1単独	6家外
保険	138057				給割 7			
記号・番号	111				22222			

氏名	ジレイ イチ 事例 ー	特記事項	
年齢	1男4平15. 5. 5 生	診療科	東京都文京区本駒込6-1-2 1 医療法人 オルカクリニック
傷	(1)左前胸骨折	診療日	() (床)

Express Zip v.2.29 © NCH Software

種類: TGZ サイズ: 80 KB 圧縮

ファイル名

- 00001-20160414-step01-01-req.txt
- 00001-20160414-step01-01-res.txt
- 00001-20160414-step02-02-req.txt
- 00001-20160414-step02-02-res.txt
- 00001-20160414-step03-03-req.txt
- 00001-20160414-step03-03-res.txt
- 00001-20160414-step04-04-req.txt
- 00001-20160414-step04-04-res.txt
- 00001-20160414-step05-05-req.txt
- 00001-20160414-step05-05-res.txt

```

00001-20160414-step02-02-res.txt - Xモ模
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
<?xml version="1.0" encoding="UTF-8"?>
<xml:io?>
<medicalv3res2 type="record">
  <Request_Number type="string">02</Request_Number>
  <Response_Number type="string">03</Response_Number>
  <Karte_Uid type="string">11</Karte_Uid>
  <Orca_Uid type="string">4641bef4-6337-40c2-b4c9-a315bec4e603</Orca_Uid>
  <Information_Date type="string">2017-02-28</Information_Date>
  <Information_Time type="string">14:49:17</Information_Time>
  <Api_Result type="string">000</Api_Result>
  <Api_Result_Message type="string">処理終了</Api_Result_Message>
  <Reskey type="string">Medical Info</Reskey>
  <Perform_Date type="string">2016-04-14</Perform_Date>
  <Department_Code type="string">11</Department_Code>
  <Department_Name type="string">整形外科</Department_Name>
  <Physician_Code type="string">10005</Physician_Code>
  <Physician_WholeName type="string">整形外科 西郎</Physician_WholeName>
  <Patient_Information type="record">
    <Patient_ID type="string">00001</Patient_ID>
    <WholeName type="string">事例 ー</WholeName>
    <WholeName_inkana type="string">ジレイ イチ</WholeName_inkana>
    <BirthDate type="string">2003-05-05</BirthDate>
    <Sex type="string">1</Sex>
  </Patient_Information>
  <HealthInsurance_Information type="record">
    <Insurance_Combination_Number type="string">0005</Insurance_Combination_Number>
    <Insurance_Combination_Rate type="string">0.3</Insurance_Combination_Rate>
    <InsuranceProvider_Class type="string">060</InsuranceProvider_Class>
    <InsuranceProvider_Number type="string">138057</InsuranceProvider_Number>
    <InsuranceProvider_WholeName type="string">国保</InsuranceProvider_WholeName>
    <HealthInsuredPerson_Symbol type="string">111</HealthInsuredPerson_Symbol>
    <HealthInsuredPerson_Number type="string">22222</HealthInsuredPerson_Number>
    <HealthInsuredPerson_Assistance type="string">3</HealthInsuredPerson_Assistance>
    <HealthInsuredPerson_Assistance_Name type="string">3割</HealthInsuredPerson_Assistance_Name>
    <RelationToInsuredPerson type="string">2</RelationToInsuredPerson>
  </HealthInsurance_Information>
  </Patient_Information>
  <Medical_Req_OffTime type="string">0</Medical_Req_OffTime>
  <Medical_Req_Information type="record">
    <Medical_Req_Info type="array">
      <Medical_Req_Info_child type="record">
        <Medical_Req_Class type="string">110</Medical_Req_Class>
        <Medication_Req_Info type="array">
          <Medication_Req_Info_child type="record">
            <Medication_Req_Code type="string">111000110</Medication_Req_Code>
          </Medication_Req_Info_child>
        </Medication_Req_Info>
      </Medical_Req_Info_child>
    </Medical_Req_Info>
  </Medical_Req_Information>
  <Medical_Req_Info_child type="record">

```

診療日	(1) 28年 4月14日	診療時間	3日
診療日	(2) 28年 4月14日	診療時間	① 日
診療日		診療時間	② 日

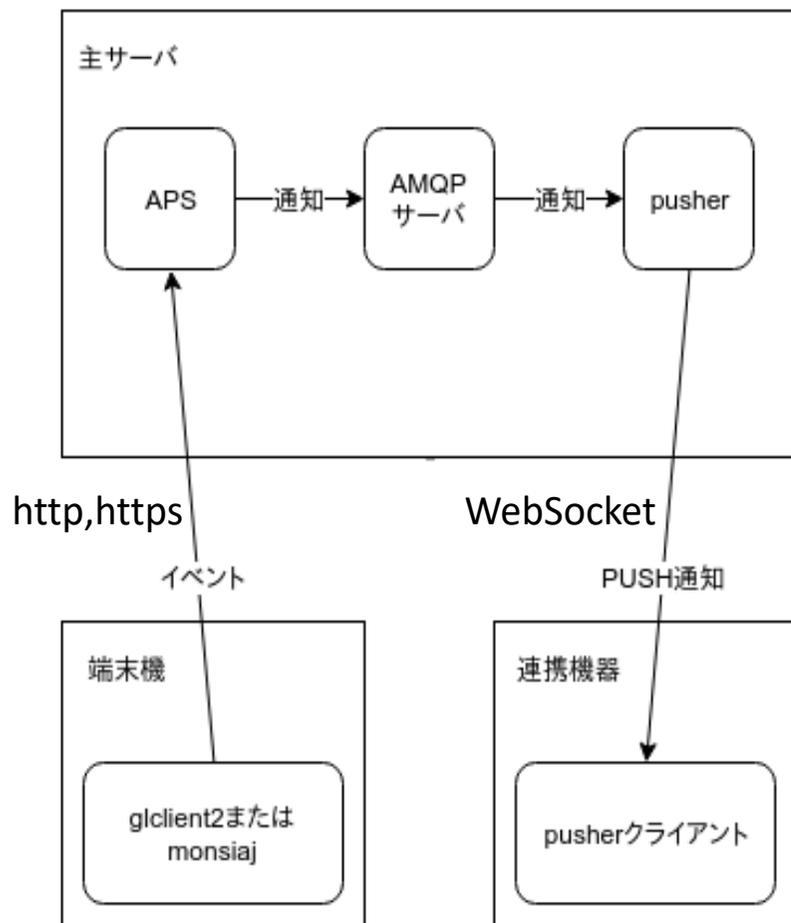
12	*明細書発行体制等加算	1 × 2
13	*薬剤情報提供料	10 × 1
21	*ケフラル種粒小児用100mg 3g 13 × 3	
22	*カロナール錠200 200mg 1錠 1 × 5	
40	*四肢ギプス包帯(半肢)(片)	780 × 1
	*創傷処置(100cm2未満)	45 × 2
	イソジン液10% 6mL	2 × 2
50	*手術 14日	
	創傷処理(筋内、臓器に達しない)(長径5cm未満)	470 × 1
	キシロカイン注射液1% 5mLV	
	生理食塩液 500mL 1瓶	20 × 1
70	*左前胸	
	単純撮影(デジタル撮影) 2回	
	単純撮影(口)の写真診断 2回	
	電子画像管理加算(単純撮影)	224 × 1
	*頭部	
	単純撮影(デジタル撮影) 2回	
	単純撮影(イ)の写真診断 2回	
	電子画像管理加算(単純撮影)	287 × 1

金額	円	公	点	公	点
※高額	円	公	点	公	点

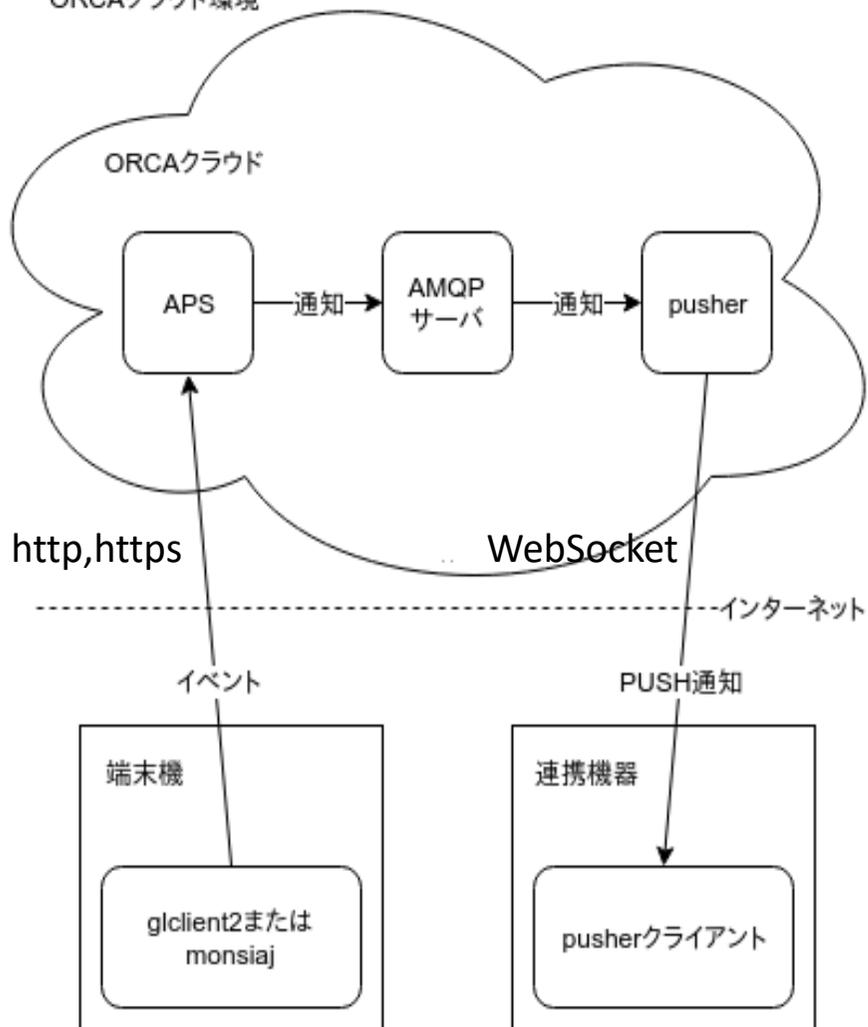


PUSH-APIの構成

オンプレ環境

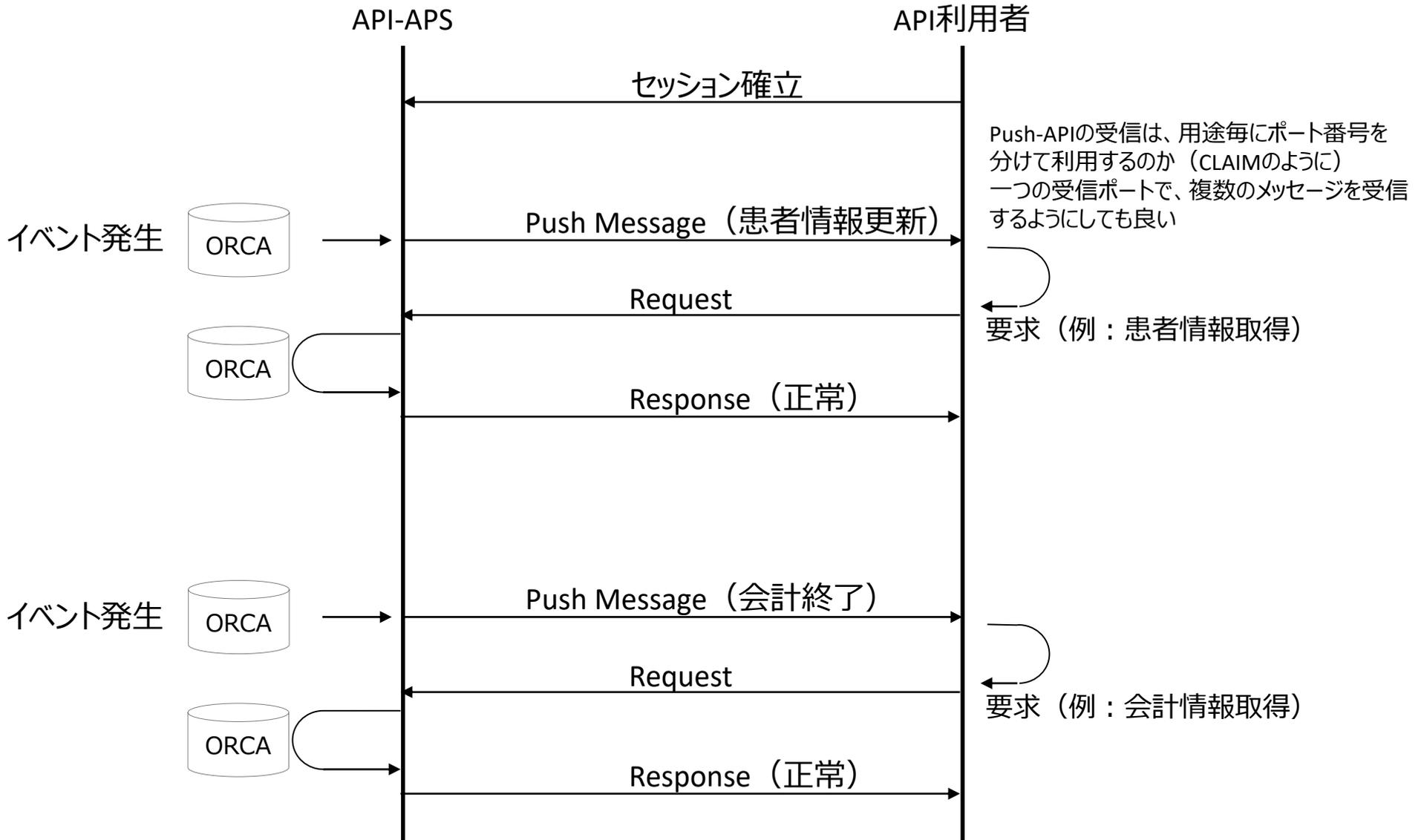


ORCAクラウド環境





PUSH-APIによる連携





PUSH – APIの使い方

イベント受信のための処理

pusher に購読するイベント名を指定する。↵

```
↵  
{↵  
  "command": "subscribe",↵  
  "req.id": "ID",          // 購読リクエストID (文字列) (リクエスト毎にユニークなID) ↵  
  "event": "patient"     // 購読するイベント名↵  
}↵
```

subscribe リクエストに対する購読成功のレスポンスメッセージである。↵

```
↵  
{↵  
  "command": "subscribed",↵  
  "req.id": "ID",          // 購読リクエストID↵  
  "sub.id": "ID"          // 購読者ID↵  
}↵
```

イベント受信

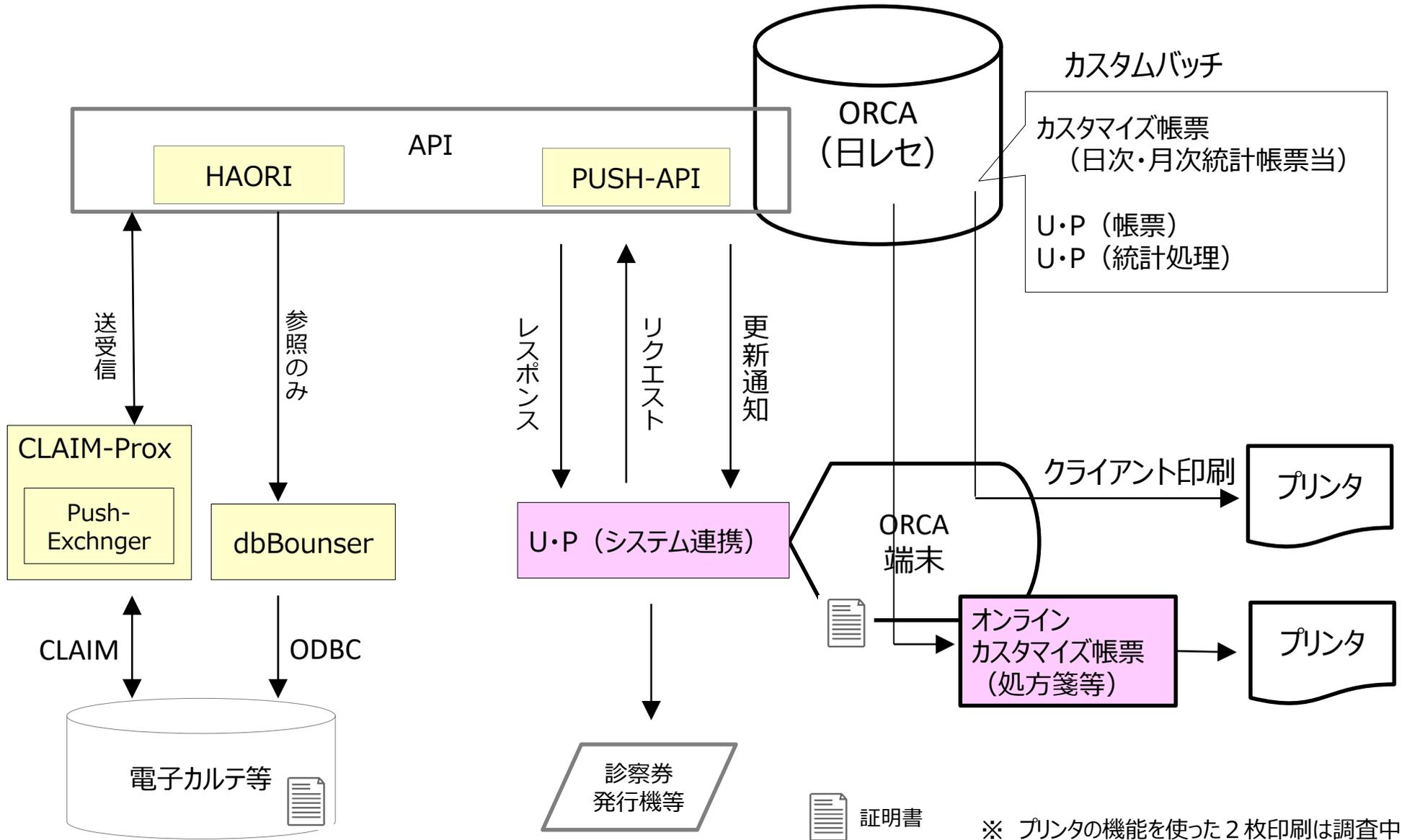
→ A P I で必要情報の取得

購読したイベントの PUSH 通知のメッセージである。↵

```
↵  
{↵  
  "command": "event",↵  
  "sub.id": "ID",          // 購読者ID↵  
  "data": {↵  
    "event": "accept",     // イベント名 ↵  
    "user": "oruser",      // イベントを発生させた日レセユーザ名↵  
    "time": "2017-01-12T14:10:17+09:00", // ISO8601 形式の日時↵  
    "body": {↵  
      ...                  // イベントによって内容は異なる↵  
    }↵  
  }↵  
}↵
```

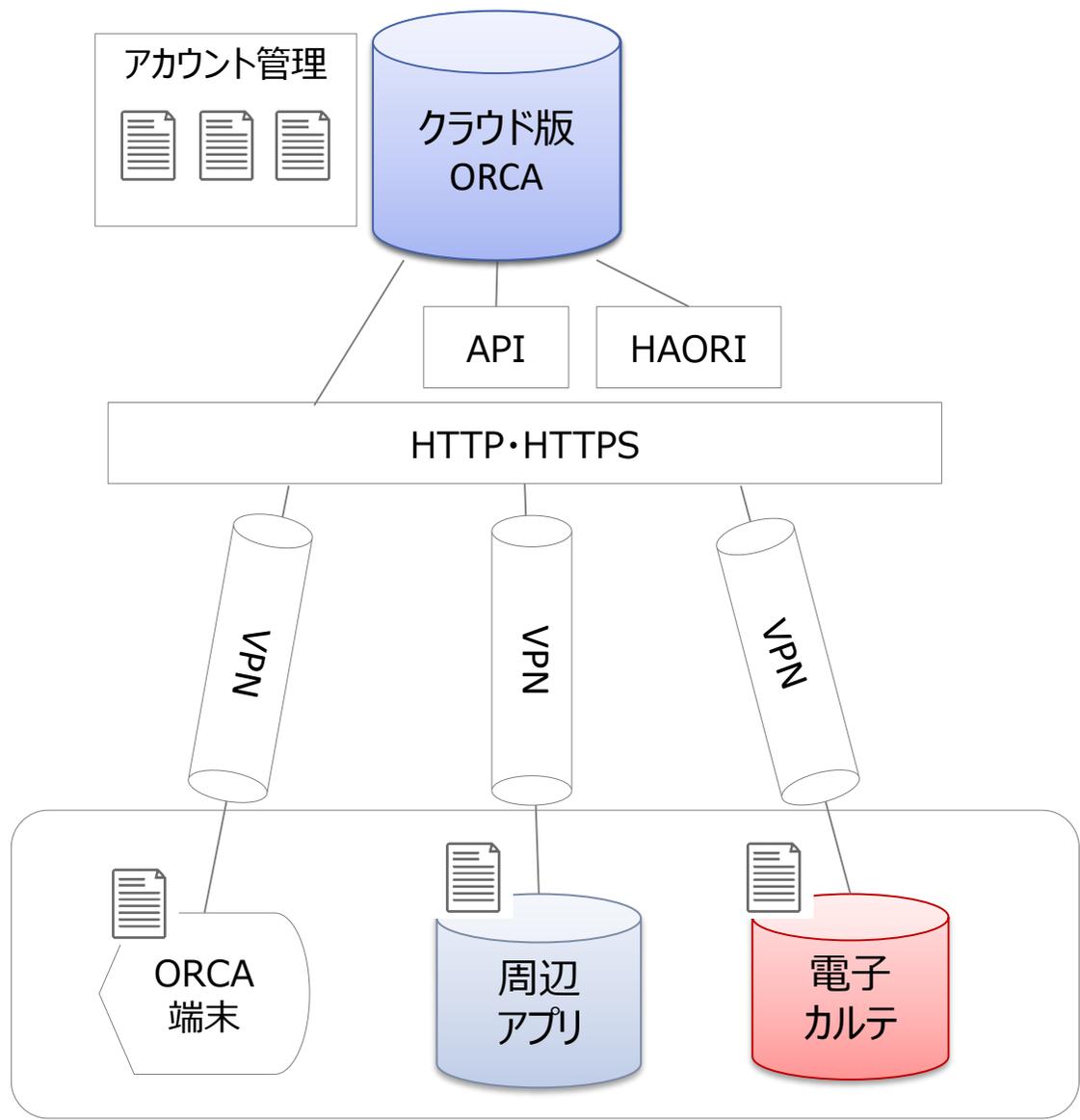


ORCAのAPI連携



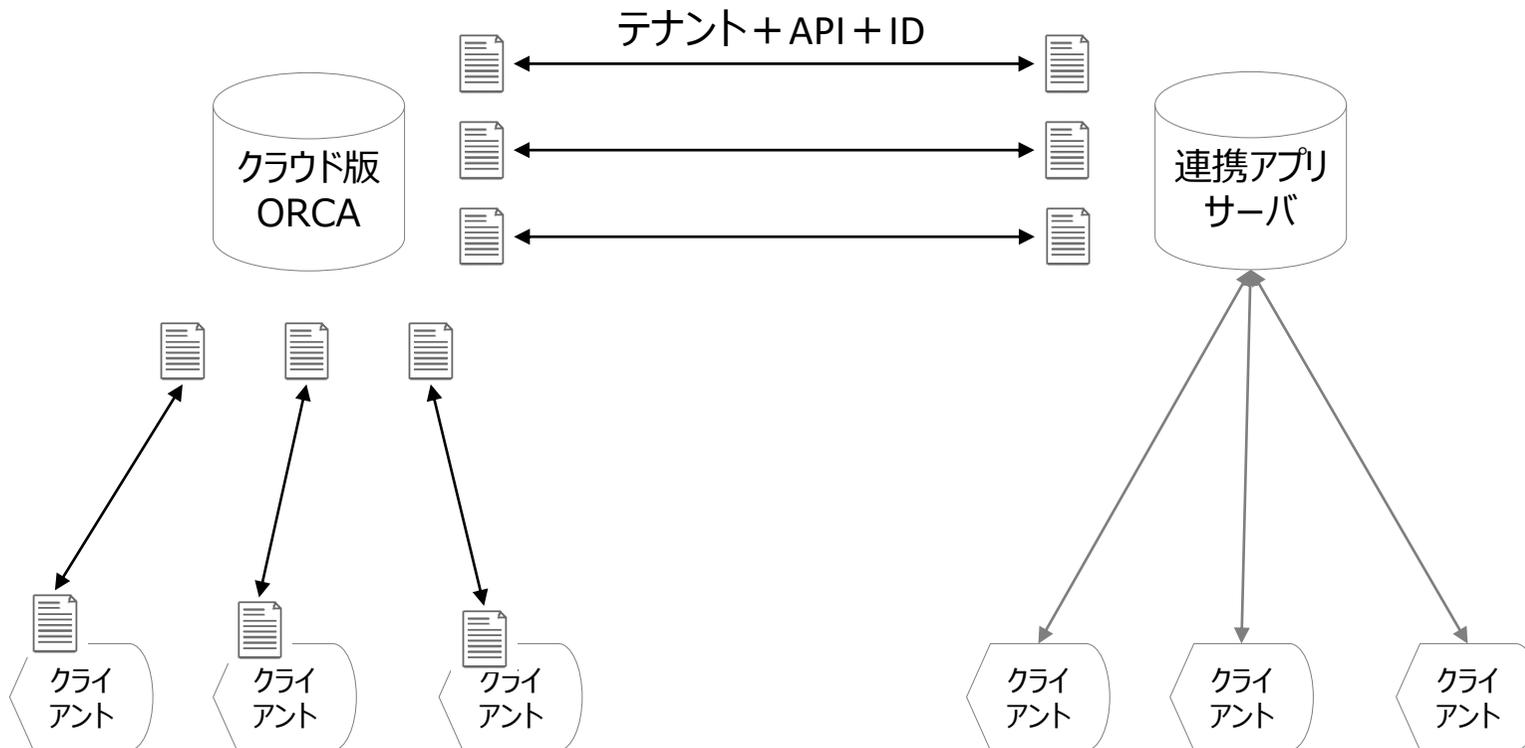


インターネットを介したAPI連携





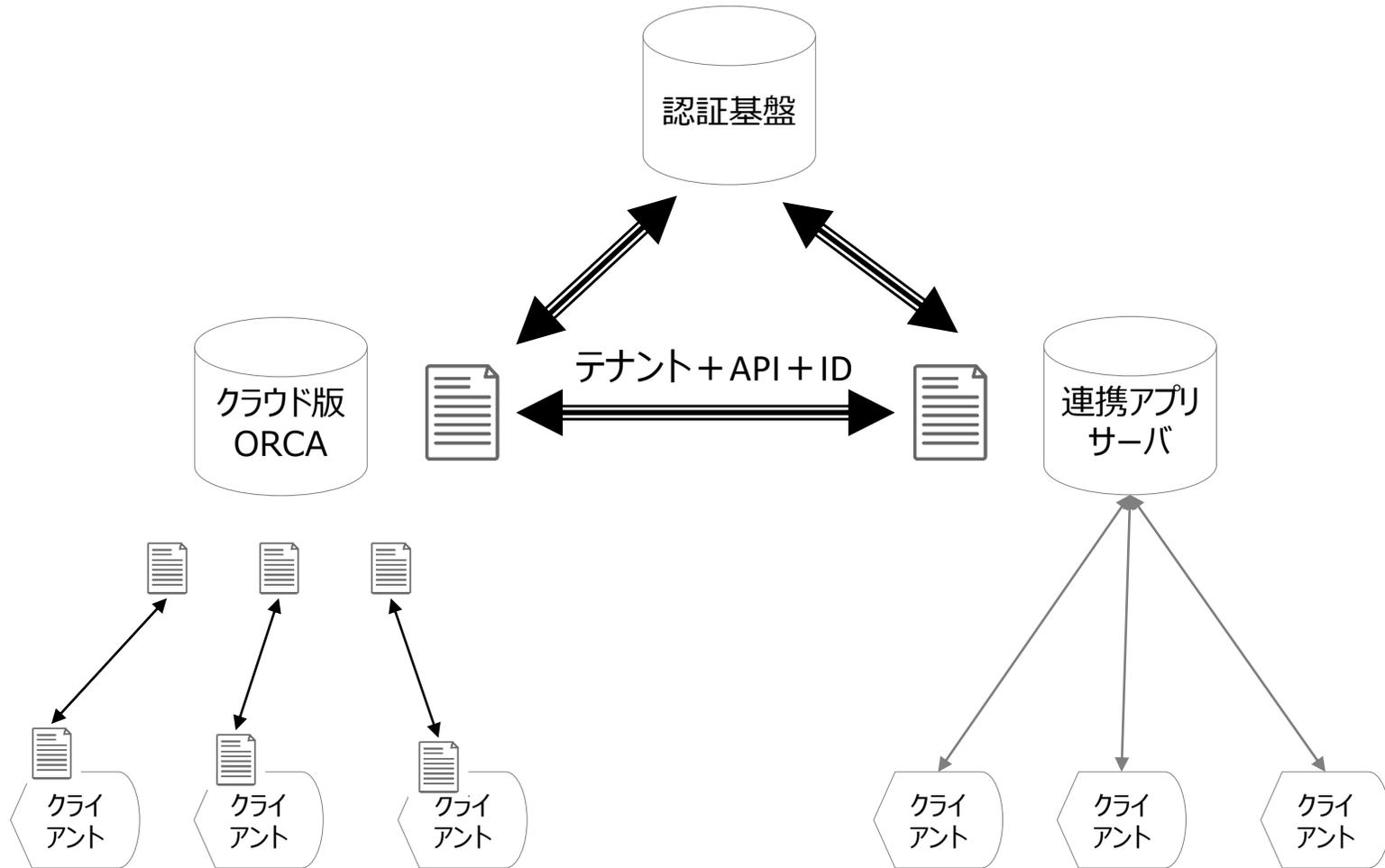
クラウドサーバ間の認証（当面の案）



各クライアント毎に証明書を登録する



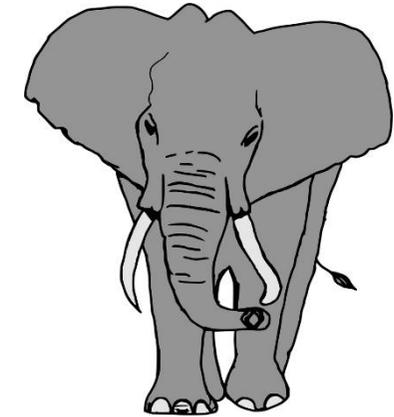
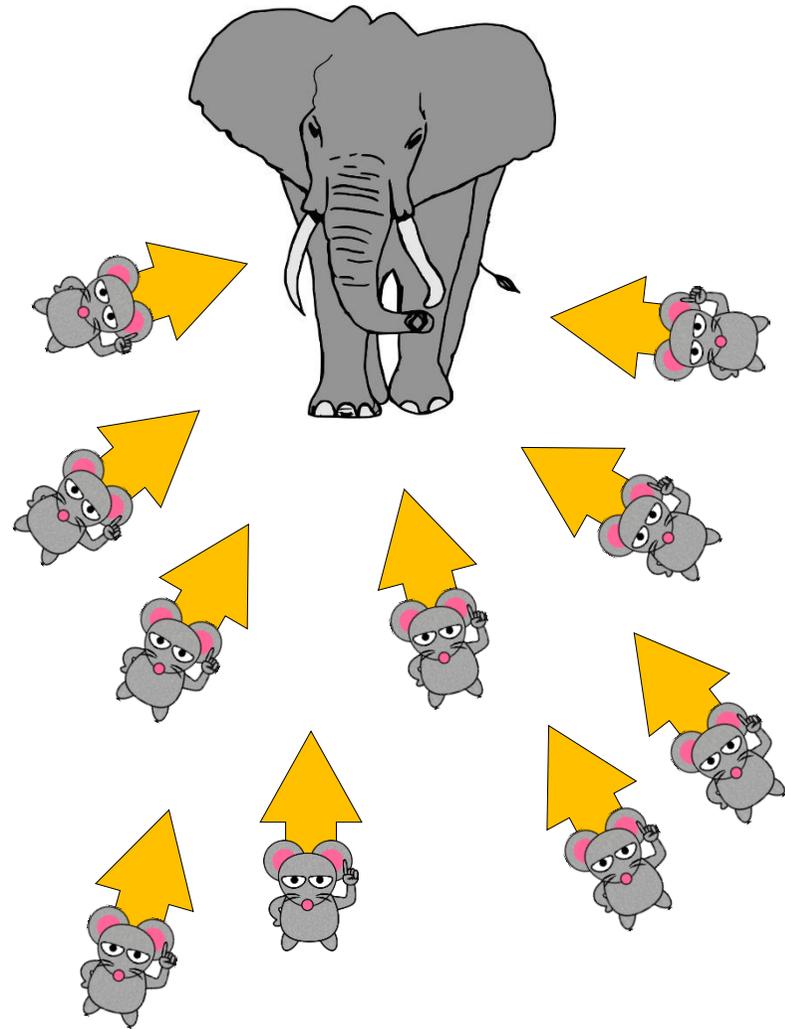
クラウドサーバ間の認証（案）



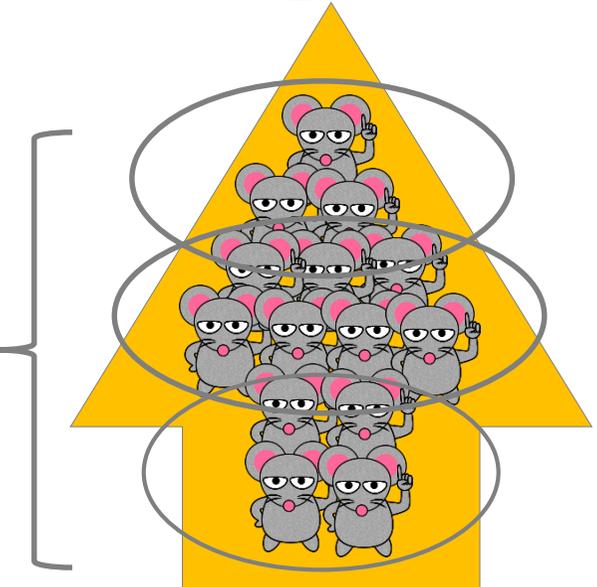
1. 認証基盤を利用
2. サーバ間で相互に認証

世の中を動かす組織力

巨象に立ち向かうには
組織力が必要！



得意分野
の集合体



ご清聴ありがとうございました。