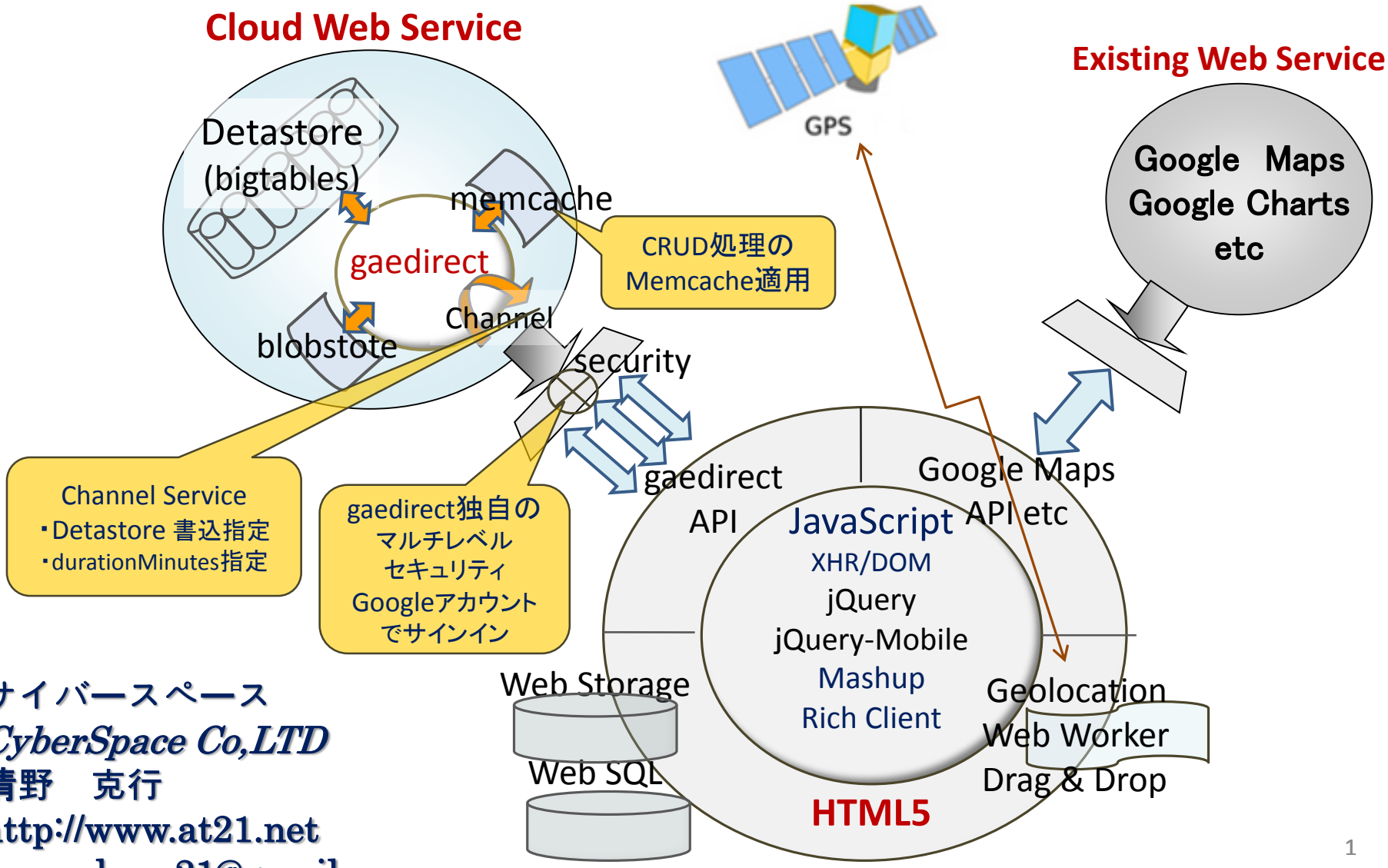
Cloud Web Service = Smart Cloud
(Java SDK 1.3.2の追加機能)

国産クラウドを開発しよう！



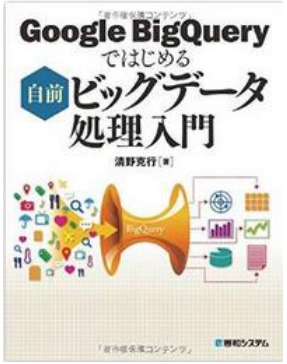
サイバースペース
CyberSpace Co,LTD
清野　克行
http://www.at21.net
superelmer21@gmail.com

自己紹介　清野　克行（セイノ　カツユキ）

superelmer21@gmail.com

日本HP、日本IBM　製造・装置業担当SE・MKTG、社内業務システム開発
ERPパッケージ、某社統合生産管理システム開発、日本語ユーティリティ
分散システム環境での引合、受注、出荷、原価、プロジェクト収支管理システム

クラウド、ビッグデータ処理、リッチクライアント、HTML5　等



@IT
atmarkIT

シンクイット
ThinkIT

CodeZine [コードジン]
開発者のための実装系Webマガジン

東南アジアで打倒アマゾン　日の丸クラウドの勝算

日本のクラウドベンダーが米国陣営に打ち勝つ。国内のクラウド市場開拓に出遅れ、アマゾン・ドット・コムなどにシェアを取られた国内ベンダーだが、実は、米国勢がそれほど浸透していない東南アジアでは、市場をリードする可能性が十分にある。アマゾンに先駆けて市場に食い込もうと、現地でデータセンターの整備に注力し始めた。

2015/5/20 6:30
日本経済新聞　電子版

■東南アジアの「一体化」、クラウドに追い風

　東南アジアでは、クラウド市場が本格的に立ち上がり、IT資源をインターネット経由で利用する需要が急速に高まっている。その背景には、東南アジア諸国連合（ASEAN）で加盟国間の「一体化」が進んでいることがある。今秋、域内貿易の自由化などを図る「ASEAN経済共同体」（AEC）が発足する予定で、これを機に、国境を越えたビジネスの活性化が期待される。

アマゾンは新規に投資してASEAN地域にデータセンターを建設すると、低コストサービスの採算が合わなくなる。そのため、データセンターの現地展開に消極的だ。そんななか、日の丸陣営の商機が高まっている。

テクニカル分野

日本はすべて後発で、追い上げトップレベルに至ったものもある

例]　自動車、家電?

〇地道な改善努力　->　トヨタ

×奇抜な発想　->　某S社　オペラ歌手を社長に抜擢

米国のICTにも弱点

・Windows8.1
・Google Chromeお気に入り、Google Maps

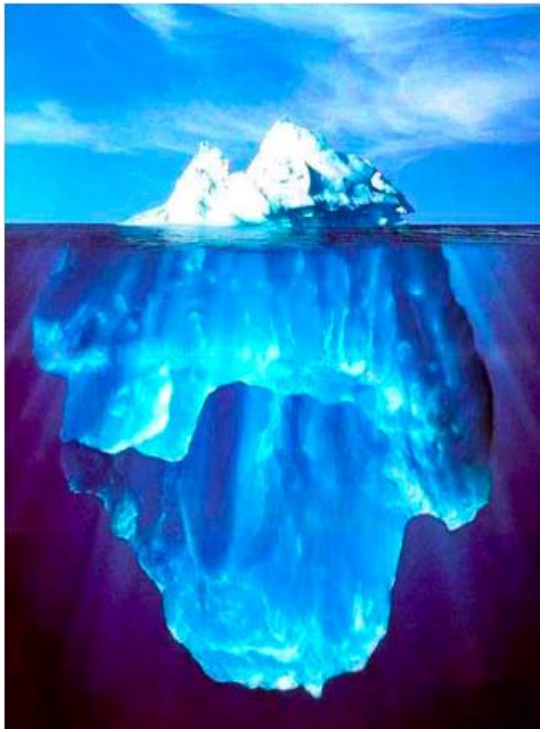
米国ICT企業から知った（学んだ？）こと

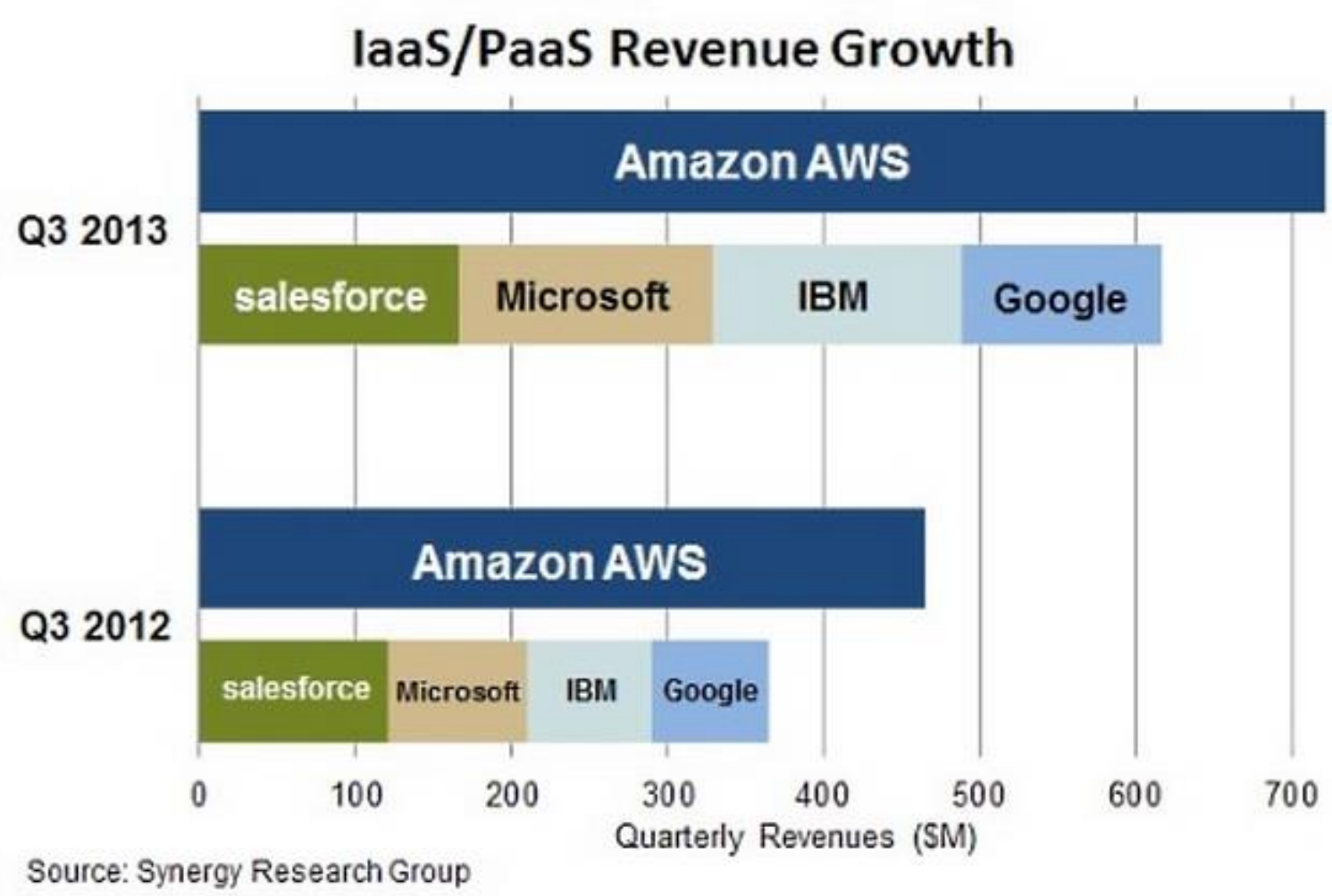
◇世に出るサービスは氷山の一角にすぎない。
◇息をするようにコードを書く。

同意する言葉

◇普通の開発者を讃えよう、人はみな並の人間だ！
http://readwrite.jp/archives/23211
◇*「変な人」でなく、みな普通の人*

◇結局、日本人は努力の総量が足りない
伊藤穰一x波頭亮
http://toyokeizai.net/articles/-/17073




IaaS/PaaS Revenue Growth
Q3 2013
Amazon AWS
salesforce
Microsoft
IBM
Google
Q3 2012
Amazon AWS
salesforce
Microsoft
IBM
Google
0
100
200
300
400
500
600
700
Quarterly Revenues ($M)
Source: Synergy Research Group

「第10回クラウドランキング」 2015/03/02
39ベストサービス/15ベストブランドを選出 （3/3）

●ベストブランドに選ばれた企業の一覧

企業名	総合スコア
グーグル	78.6
セールスフォース･ドットコム	78.1
日本IBM	75.7
日本マイクロソフト	74.6
アマゾン･ドット･コム	74.4
ヴイエムウェア	72.4
NTTデータ	71.2
日本オラクル	69.7
NTTコミュニケーションズ	67.5
富士通	67.2
NEC	66.7
インターネットイニシアティブ（IIJ）	66.6
ヤフー	64.1
日立製作所	63.0
シスコシステムズ	62.6

表1●ベストサービスに選ばれた企業とサービスの一覧

企業名（サービス名）	総合スコア
クラウド基盤サービス（IaaS/PaaS）部門	
日本マイクロソフト［Microsoft Azure］	66.8
日本IBM［IBM SoftLayer］	66.6
ニフティ［ニフティクラウド］	66.2
NTTコミュニケーションズ［Bizホスティング］	65.9
富士通［FUJITSU Cloud IaaS Trusted Public S5］	65.1
ビッグローブ［BIGLOBEクラウドホスティング］	64.8
日本ユニシスグループ［U-Cloud IaaS Select］	64.3
インターネットイニシアティブ（IIJ）［IIJ GIOサービス］	63.5
IDCフロンティア［IDCフロンティア クラウドサービス］	63.0

ニフティクラウド

サーバ仮想化の延長？

ディスク

Replicationとその方法は？

・HRD、Master/Slave

Linuxのビューション　CentoOS, Debian, Red Hat, SUSE, Ubuntu

MBaaS

・データストアアクセスがない？

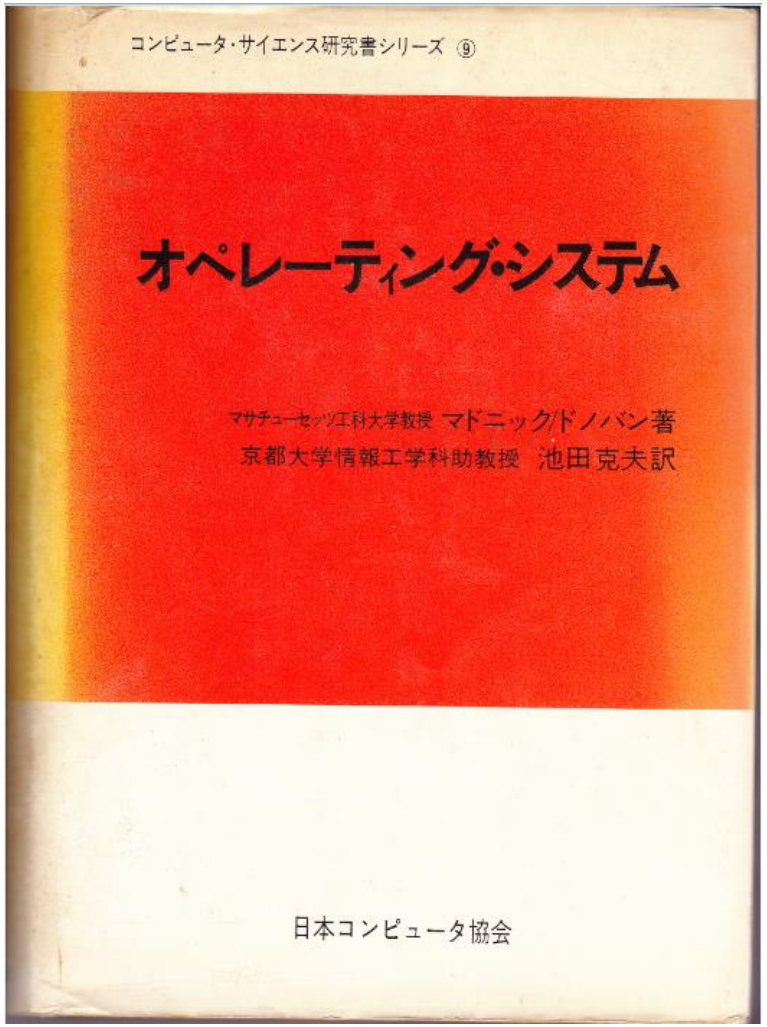
->BigTableをアクセスできるgaedirect参照

Bizホスティング

NTTコミュニケーションズのデータセンターに設置された仮想サーバーを提供する、クラウド型仮想ホスティングサービスです。
仮想化されたサーバーリソースを、VPN直結のネットワーク経由でセキュアに、かつオンデマンドに提供。コスト削減・運用の利便性・強力なセキュリティ、企業が次世代ICT環境・クラウドに望むすべてを兼ね備えたサービス

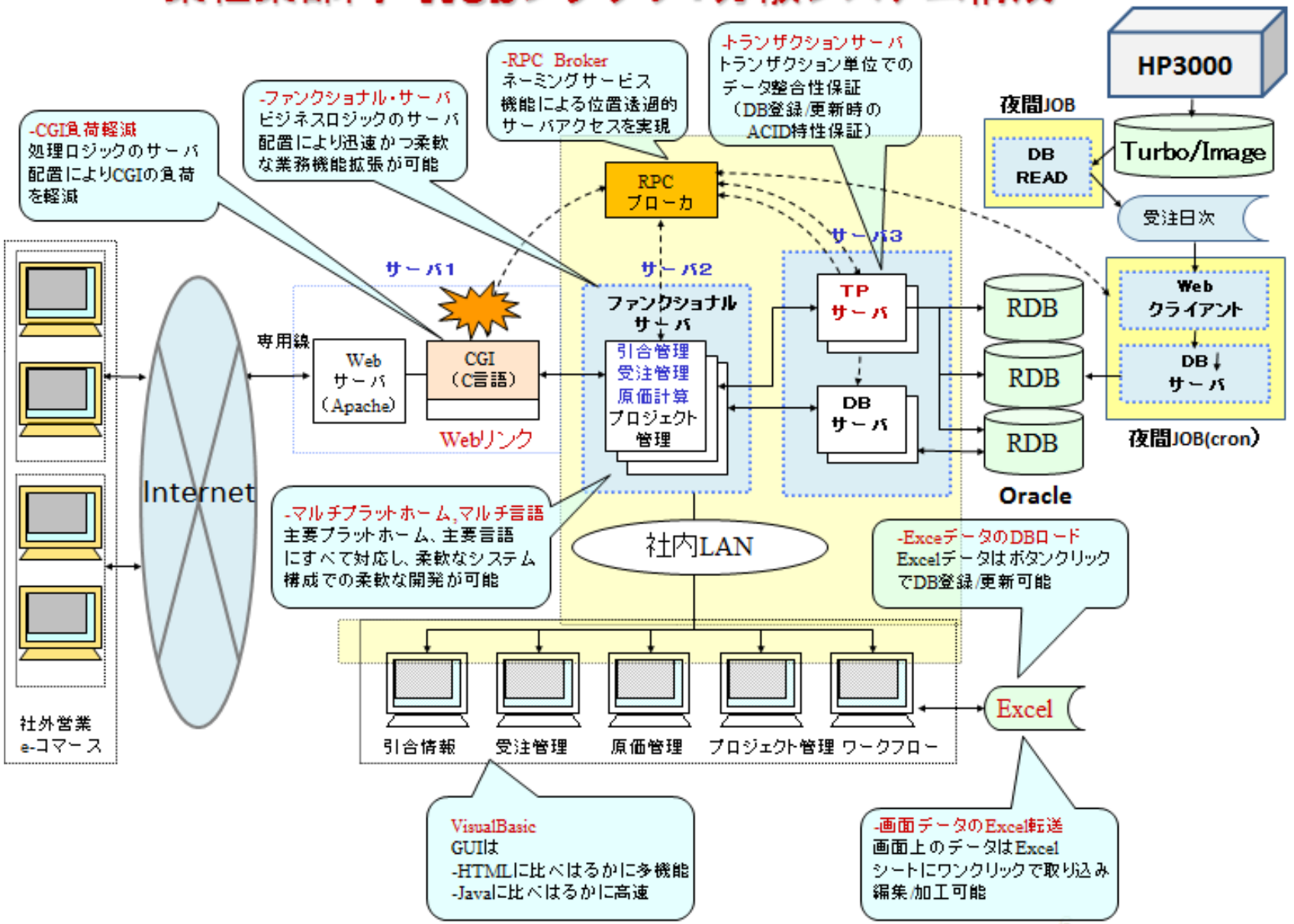
John J. Donovan

MIT教授
IBMメインフレームOSの設計・開発主導
博士号4　内　理系2、文系2



バックグラウンド：ジョン・ドノバンが考案した分散システム上での開発

某社某部門 Webブラウザ+分散システム構成



Next Cloud

国産オリジナルクラウドの構成

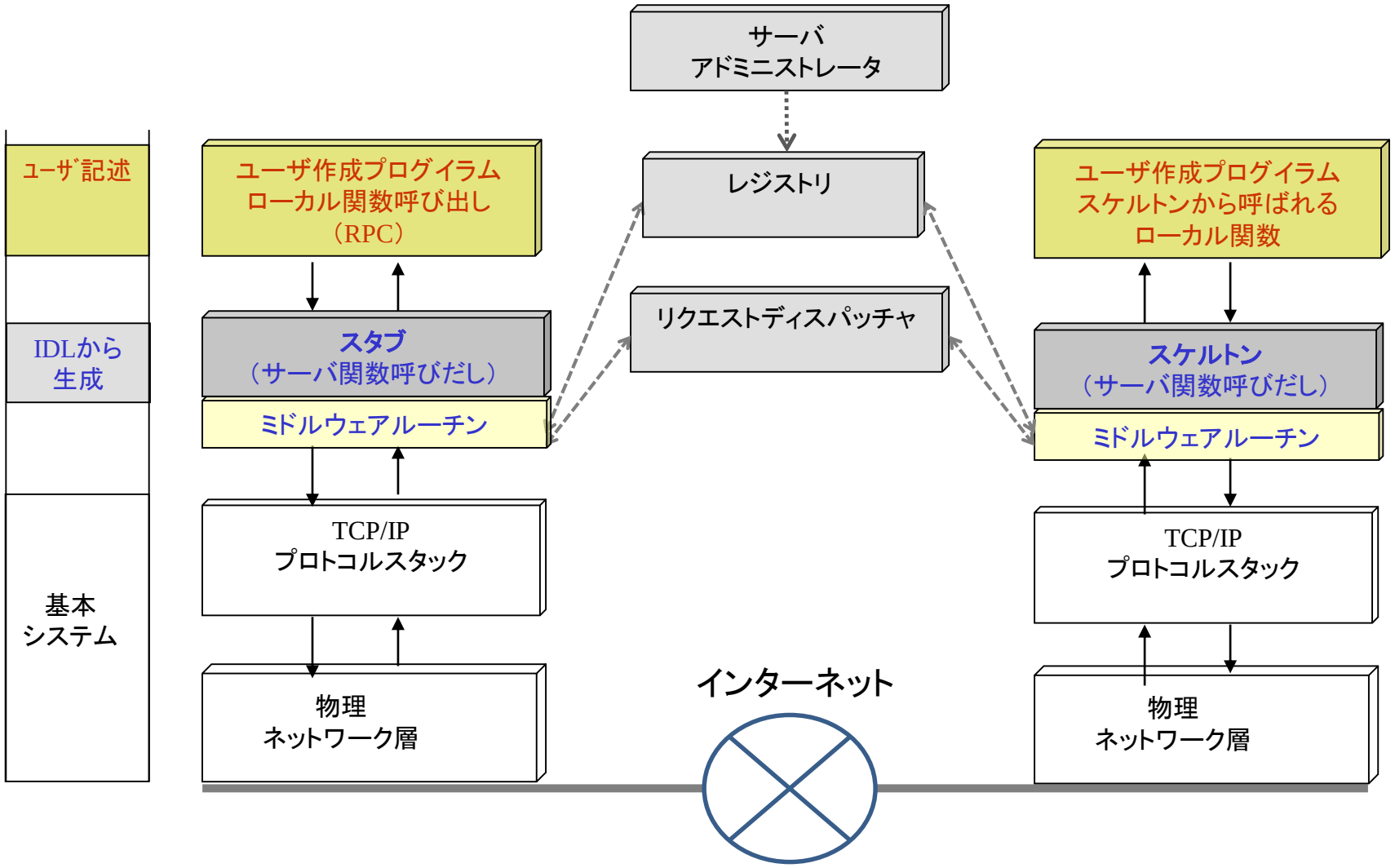
1．ハイブリッドクラウド
　無理のない導入
　段階的移行
2．PaaSベース
　巨大分散システム　<= CORBA/IIOP, OSF/DCE
　基本仮想化を使用しない
　◇ベアメタルサーバでサーバ能力１００％利用
　◇コンテナ上PaaSでユーザのシステム構成対応

・PaaSはIaaSにくらべユーザ（開発・保守）の負担が極端に少なく、
　アプリ開発の生産性が遥かに高い。
　Google：　App Engine vs Compute Engine

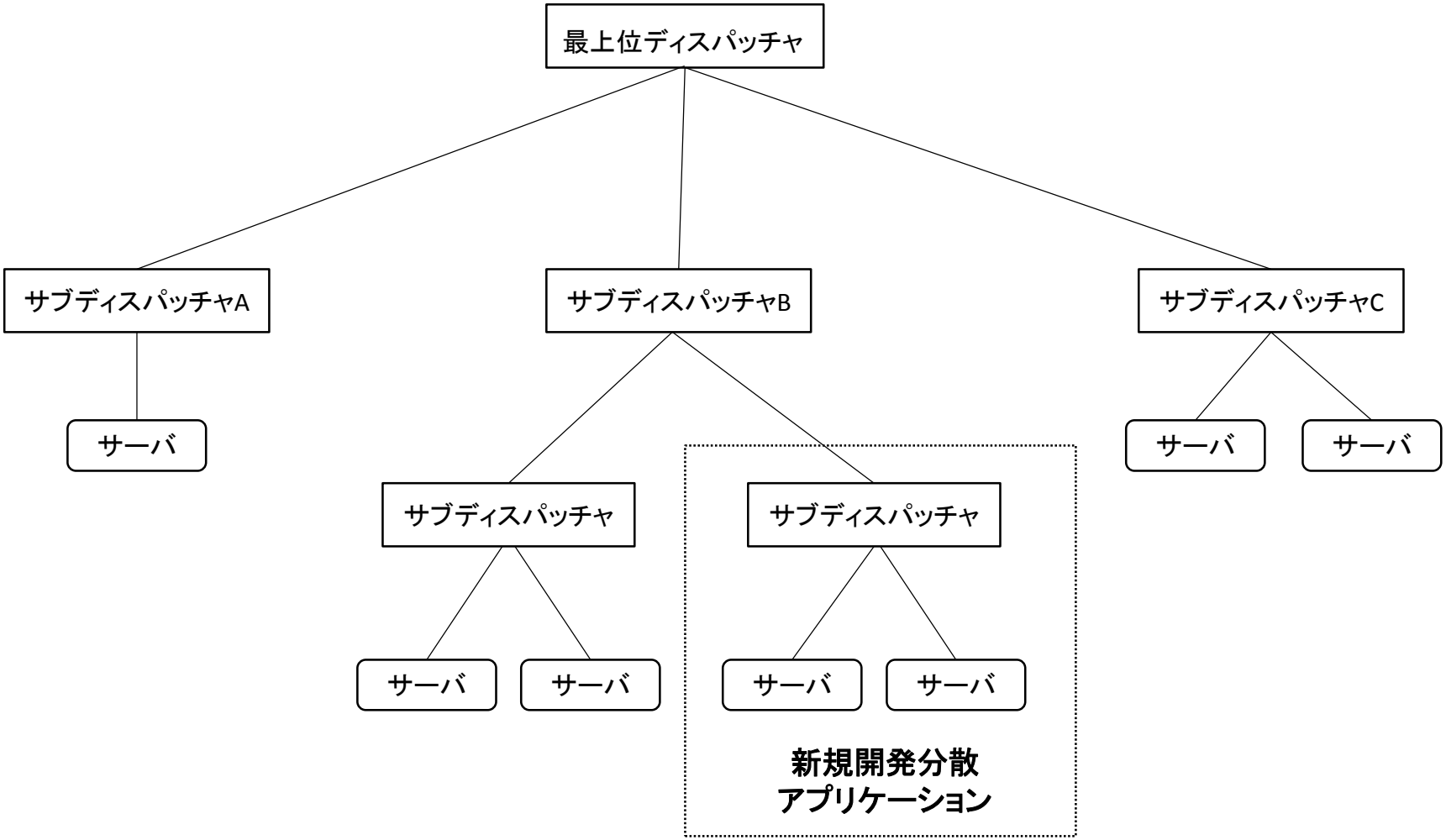
Web(AP)サーバ　+　主要言語　+　主要RDB　(+ NoSQL)

分散システムの物理構成

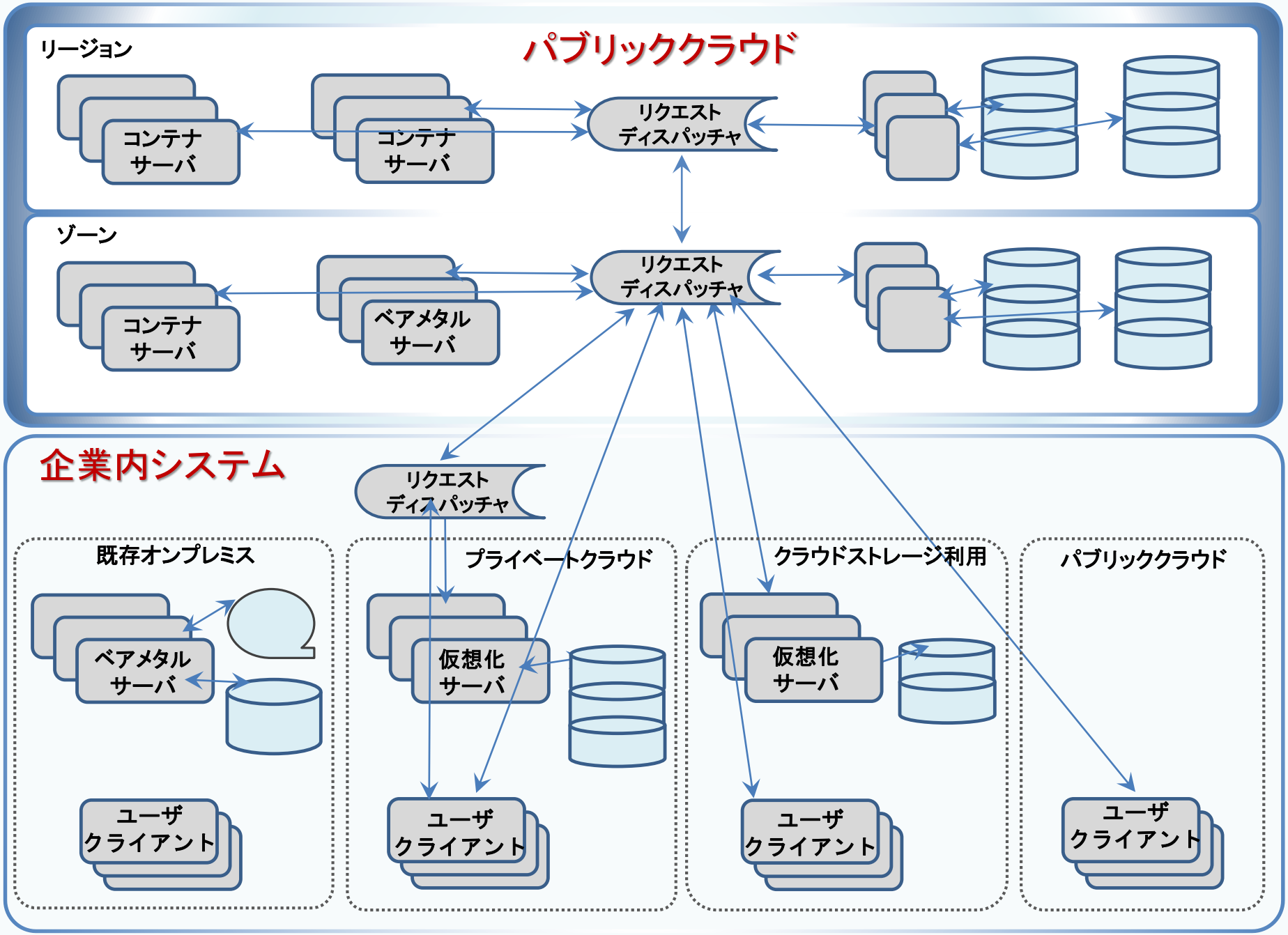
PaaS分散システムの物理構成では、URLのRESTアクセスから後は、トランスポート層上のSocket通信により高速データ通信が可能



PaaS分散環境でのサービスリクエストは（リクエスト）ディスパッチャが処理サーバの割り当てを行うが、ディスパッチャの階層構造によって、
・オンプレミス環境とパブリッククラウド連携
・プライベートクラウドのパブリッククライド移行
をスムーズに行う事が出来る。



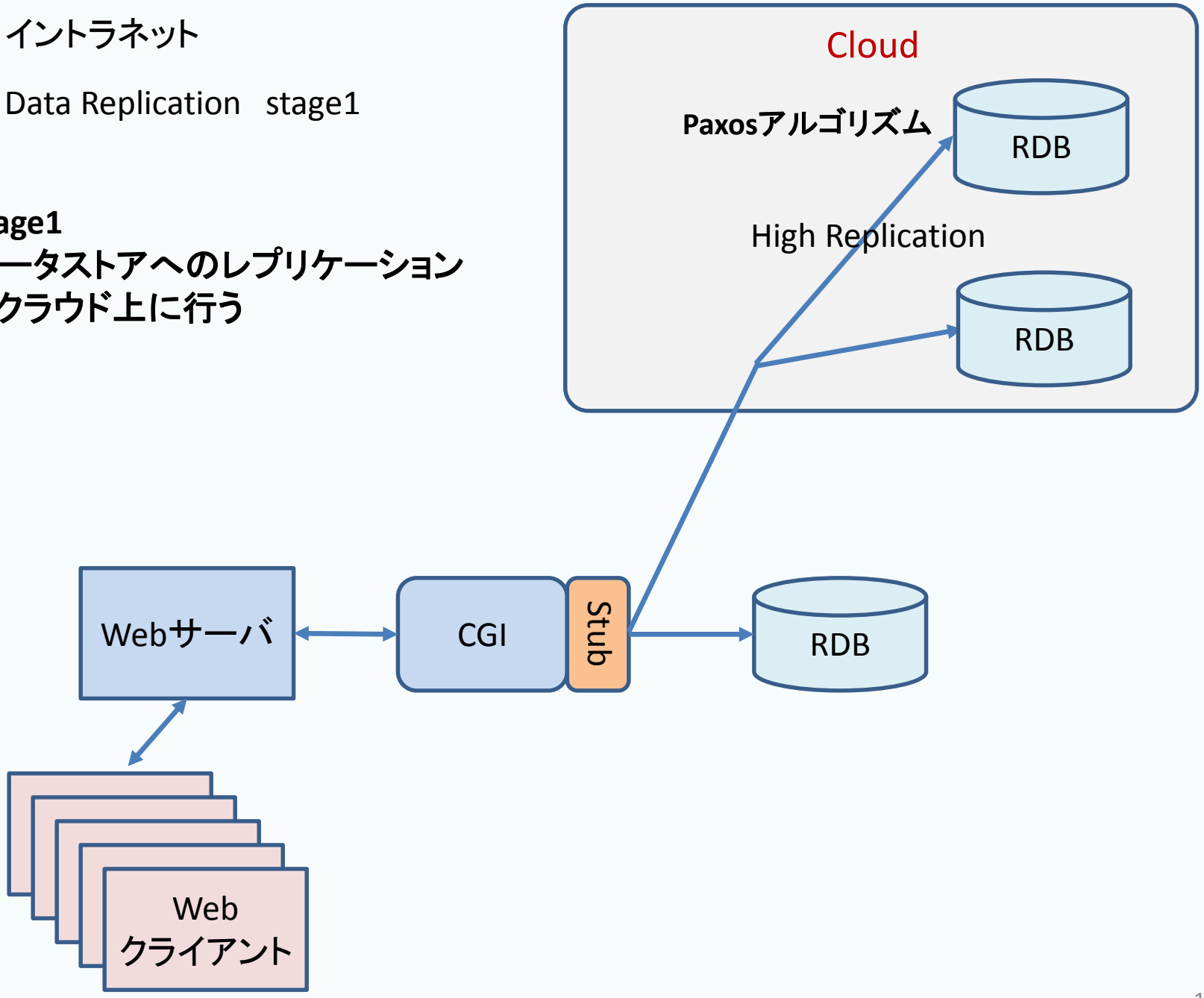
リクエストディスパッチャを使用したオンプレミスからパブリッククラウドへの移行例



イントラネット

Data Replication　stage1

Stage1
データストアへのレプリケーション
をクラウド上に行う

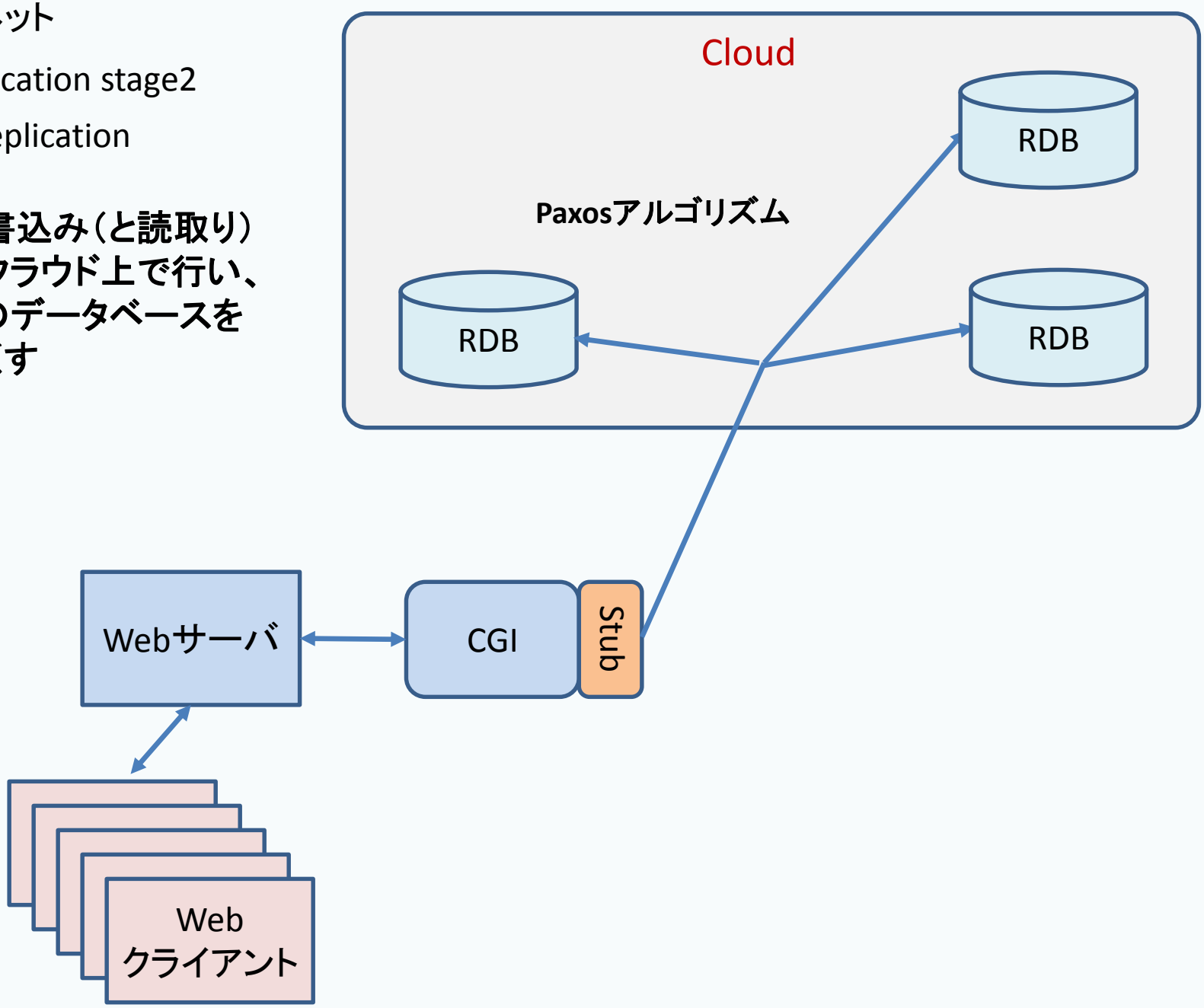


イントラネット

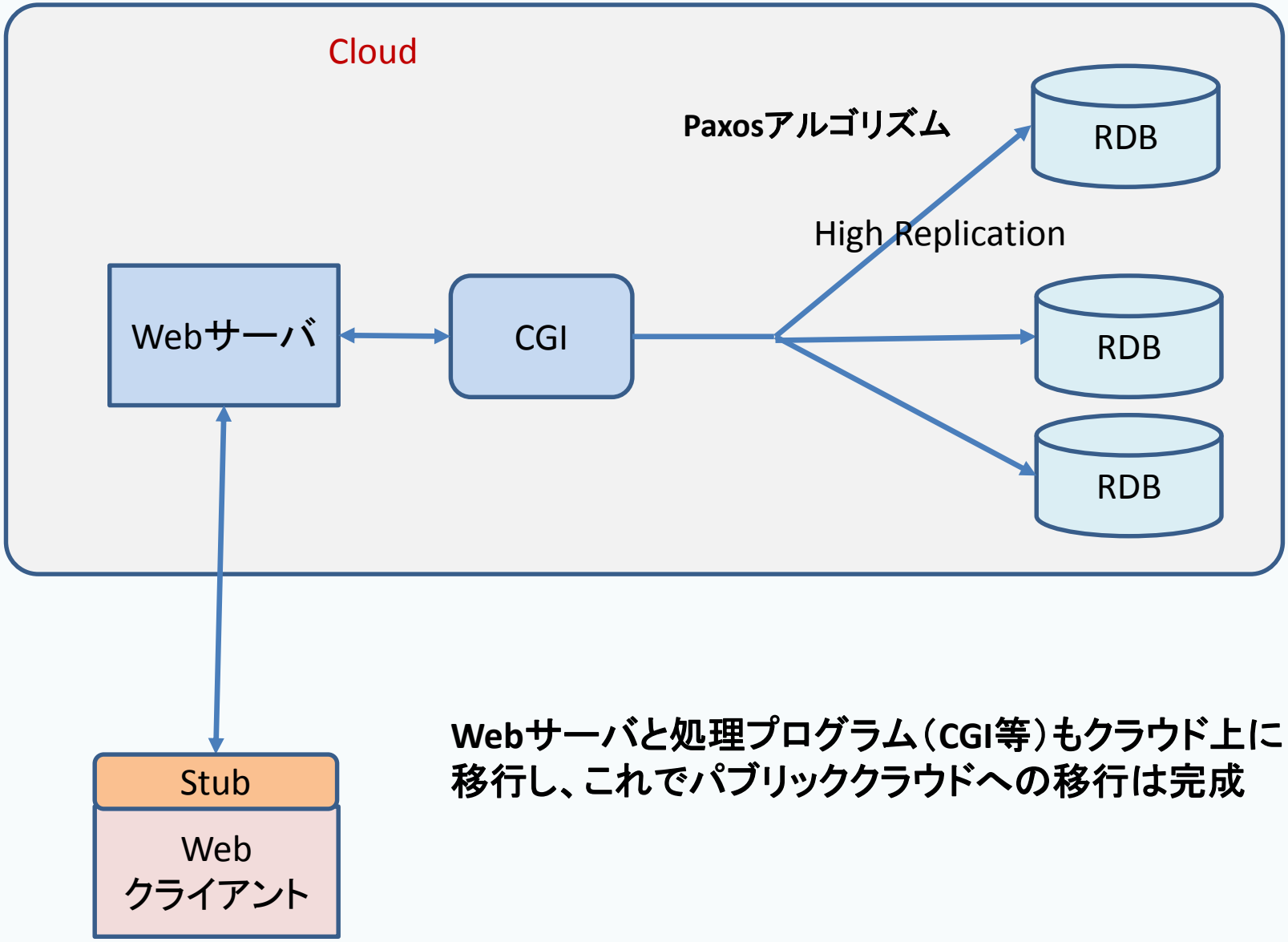
Data Replication stage2

High Replication

データの書込み（と読取り）をすべてクラウド上で行い、オンプレのデータベースをすべてなくす

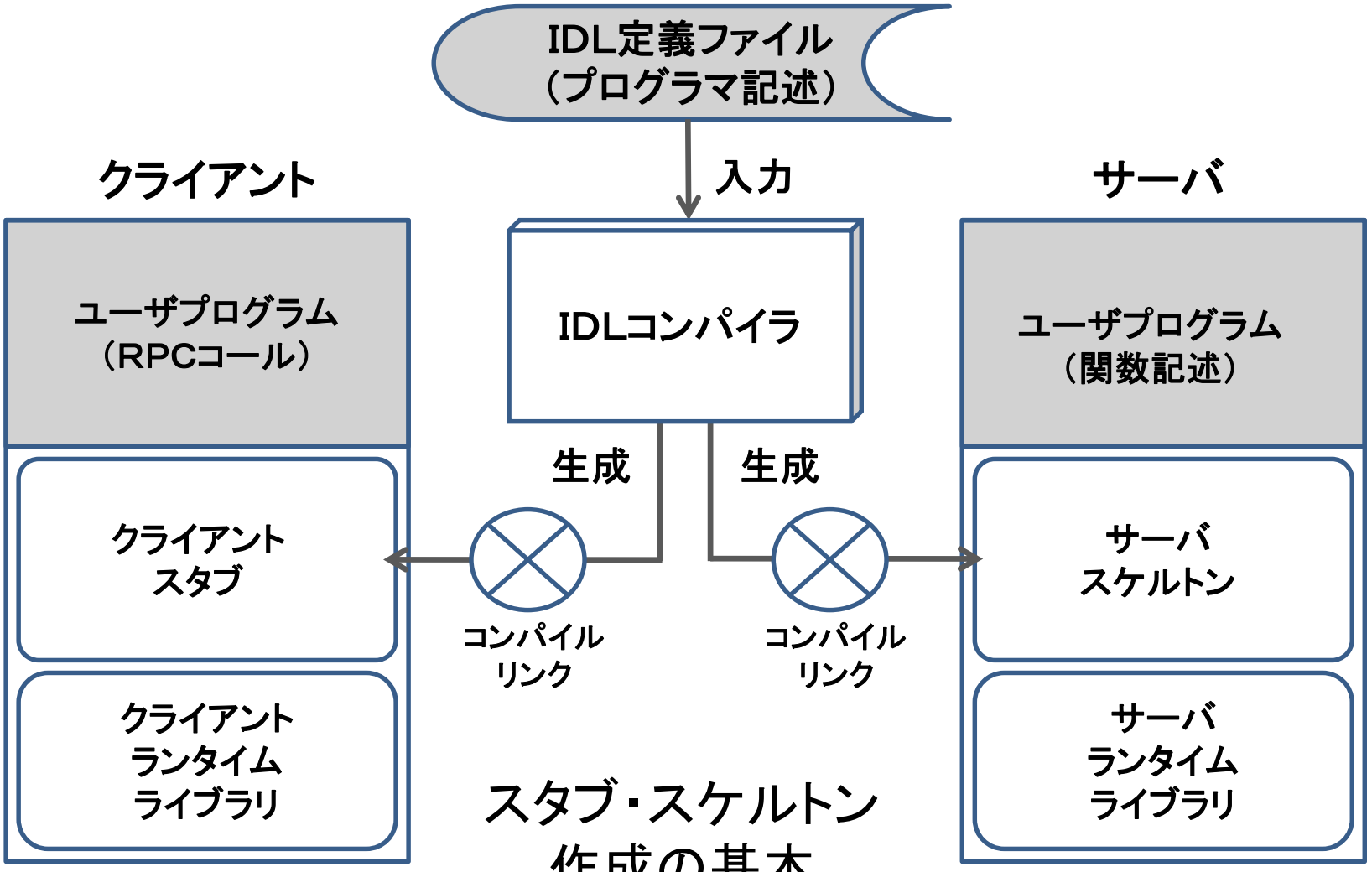


イントラネット　stage3



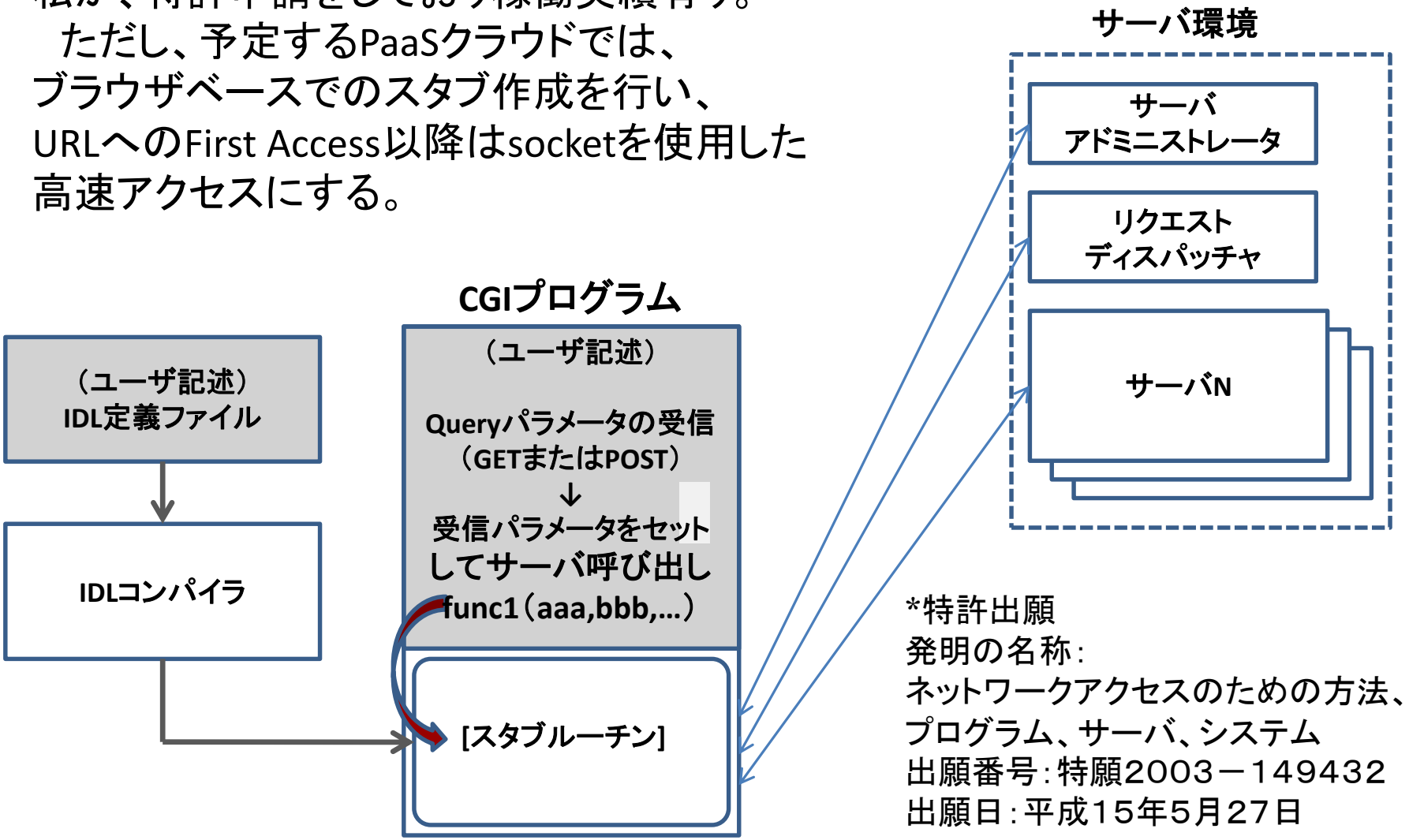
Webサーバと処理プログラム（CGI等）もクラウド上に移行し、これでパブリッククラウドへの移行は完成

ところで、データストアCRUDのAPIは下記IDLから生成されるスタブがスティールして行い、レプリケーションデータをクラウドへ書き込むような変更で
プログラム記述の変更は、ごくわずかで、かつパターン化された内容で可能

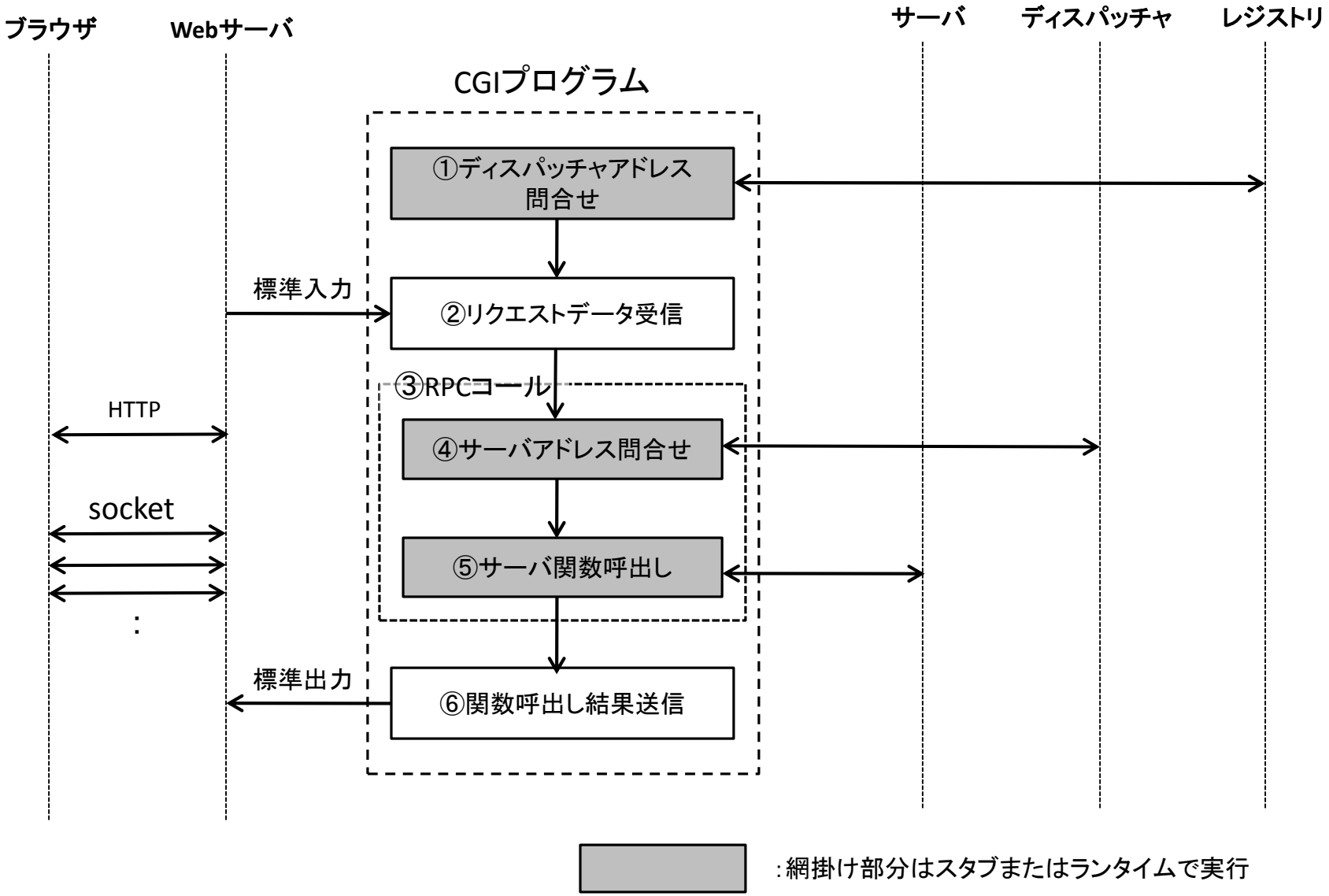


スタブ・スケルトン
作成の基本

CGIを分散環境に組み込みリクエストディスパチャの支配下で稼働させる方式は私が、特許申請をしており稼働実績有り。
　ただし、予定するPaaSクラウドでは、
ブラウザベースでのスタブ作成を行い、
URLへのFirst Access以降はsocketを使用した
高速アクセスにする。

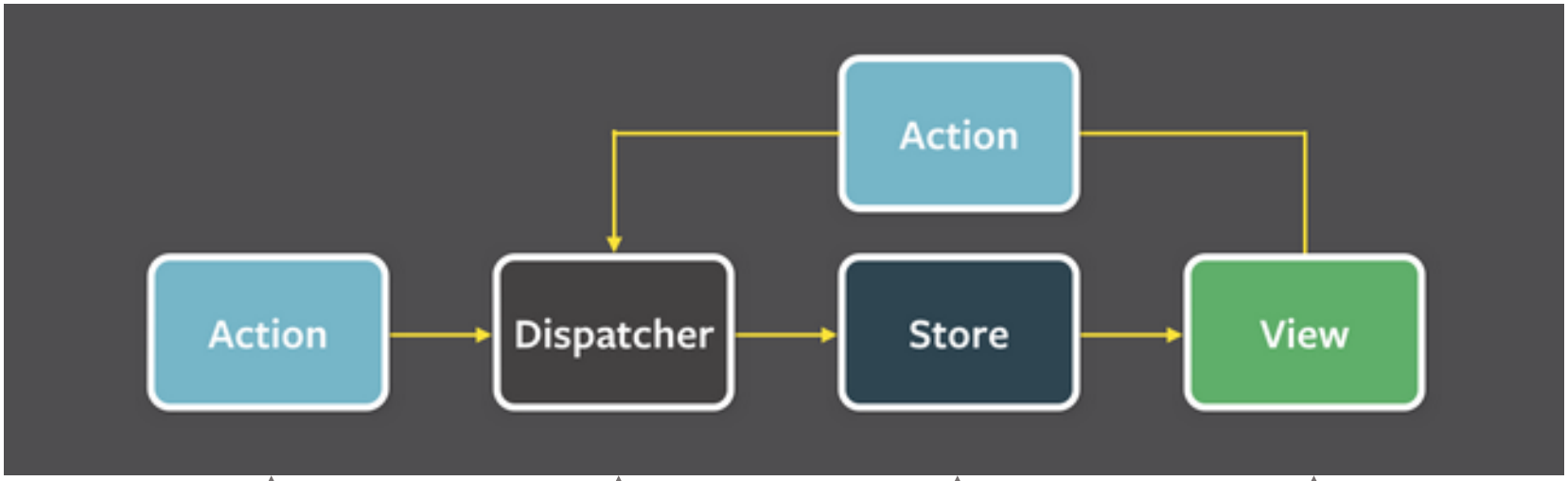


*特許出願
発明の名称：
ネットワークアクセスのための方法、
プログラム、サーバ、システム
出願番号：特願2003－149432
出願日：平成15年5月27日


ブラウザ
Webサーバ
サーバ
ディスパッチャ
レジストリ
CGIプログラム
①ディスパッチャアドレス
問合せ
標準入力
②リクエストデータ受信
③RPCコール
HTTP
④サーバアドレス問合せ
socket
⑤サーバ関数呼出し
標準出力
⑥関数呼出し結果送信
:網掛け部分はスタブまたはランタイムで実行

Facebook　Parseも分散システムベース

分散システムベースのPaaS型クラウドであれば、FacebookのParseのような、ユーザプログラム作成環境を容易につくることができる。



Reactで
統一パターン

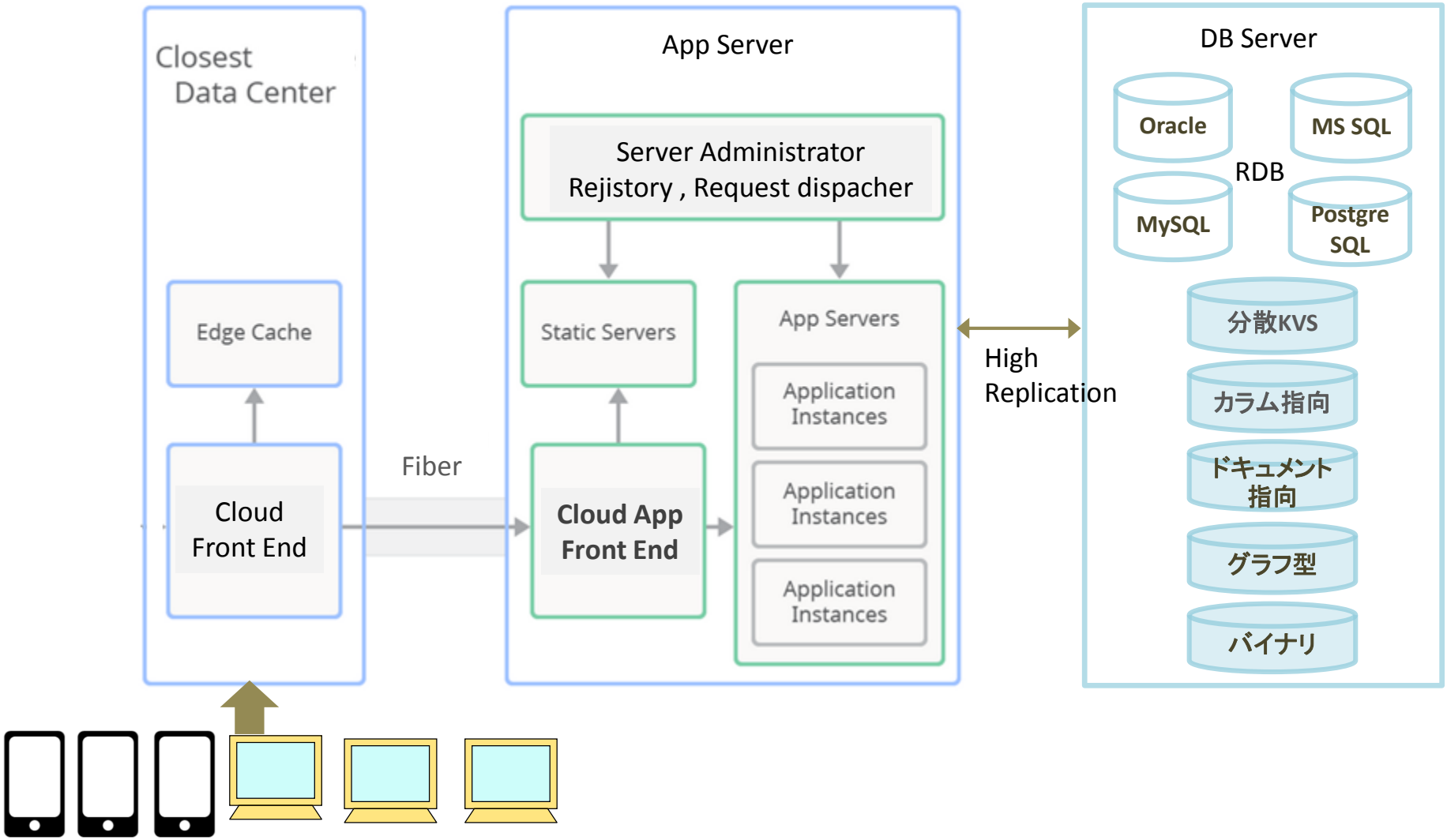
Request Dispacher
で巨大分散環境

NoSQLで
FBユーザ数
リクエスト数
対応

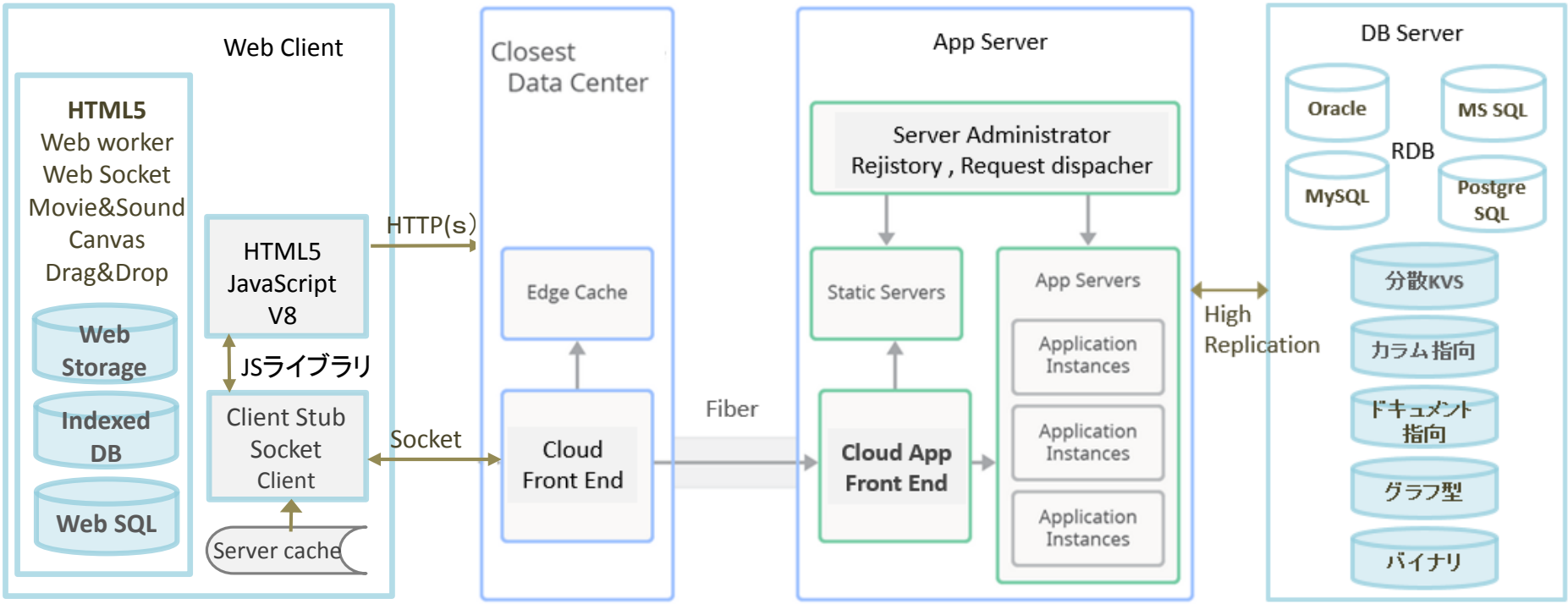
Reactで
統一パターン

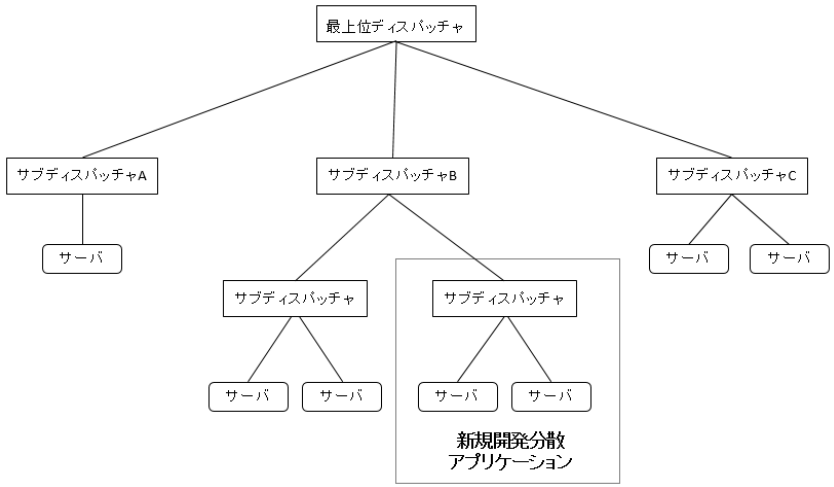
以上の前提によるPaasSクラウドのサーバ構成

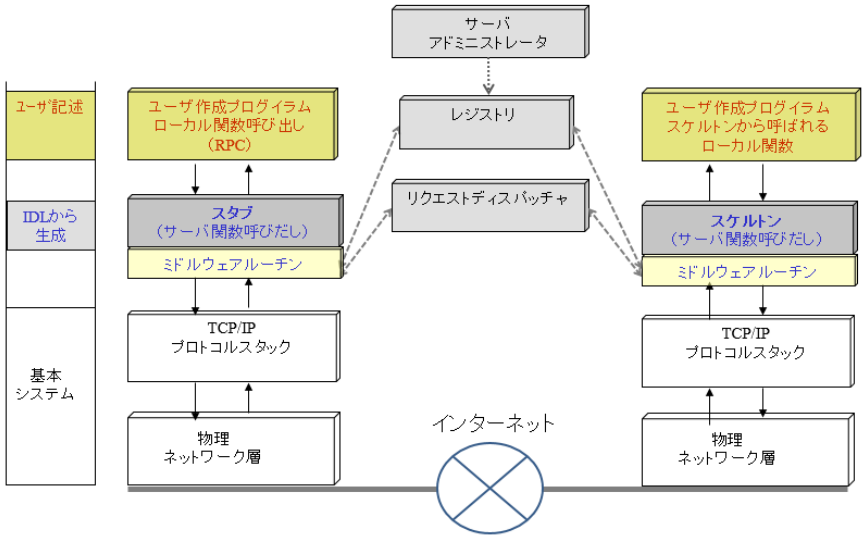
PaaSではGoogleのApp Engineが良く知られているが、App EngineはRDBサポートがMySQLのみという弱点がある、これをAmazon　AWS等々のOracle,MS SQL Server、MySQL,PostgreSQLサポートにレベルアップし、またNoSQLも分散KVSの他、カラム指向ドキュメント指向、グラフ型をサポートしまた画像、音声、動画用のバイナリデータサポートを追加する。
NoSQLはすべてオープンソースで構成する。



Next Cloud Total formation



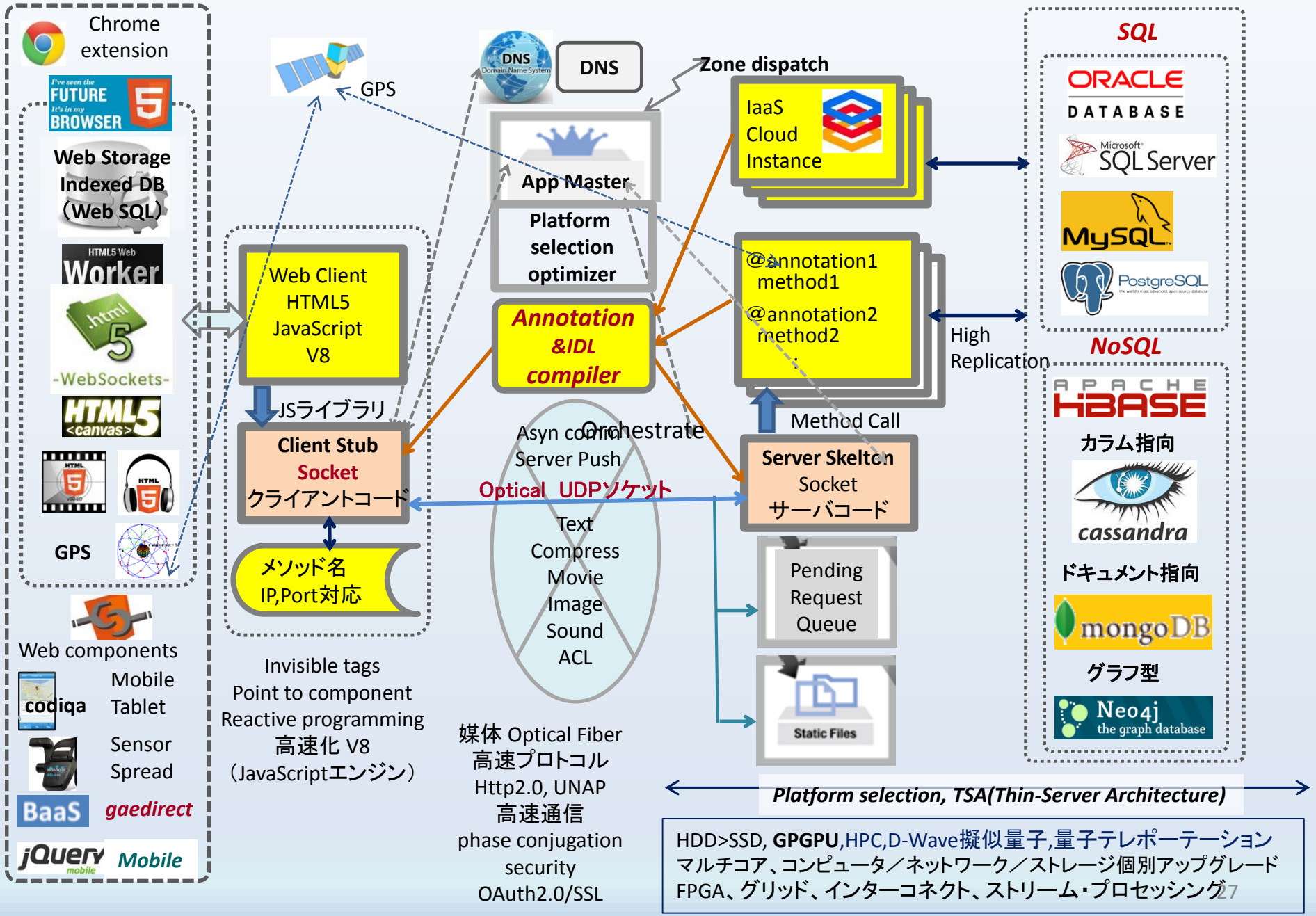




Next Cloud with Total Distributed System

K.Seino

・Cloud First, Offline First, Mobile First, Real time Web, Client MVC

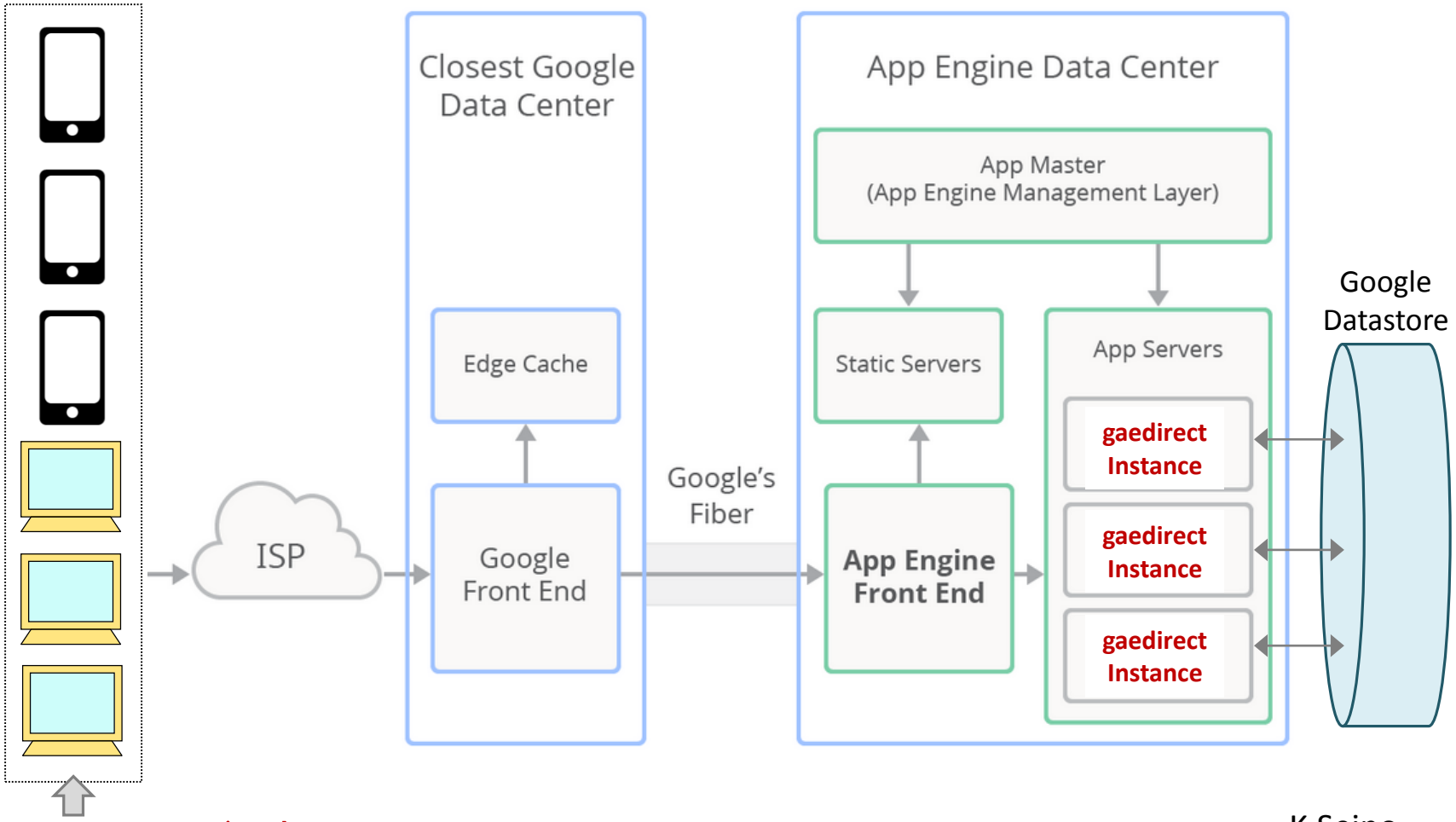


HDD>SSD, **GPGPU**,HPC,D-Wave擬似量子,量子テレポーテーション
マルチコア、コンピュータ／ネットワーク／ストレージ個別アップグレード
FPGA、グリッド、インターコネクト、ストリーム・プロセッシング

MBaaS

BaaS・MBaaS ツール　gaedirect

Google App Engineの処理をサーバ側コード記述を行わず、JavaScript（jQuery）のみで行える！
・Google Cloud Datastoreの登録・参照・更新・削除処理、および条件検索
・Channelサービスを使用した、サーバプッシュ処理
・blobstoreの参照・表示

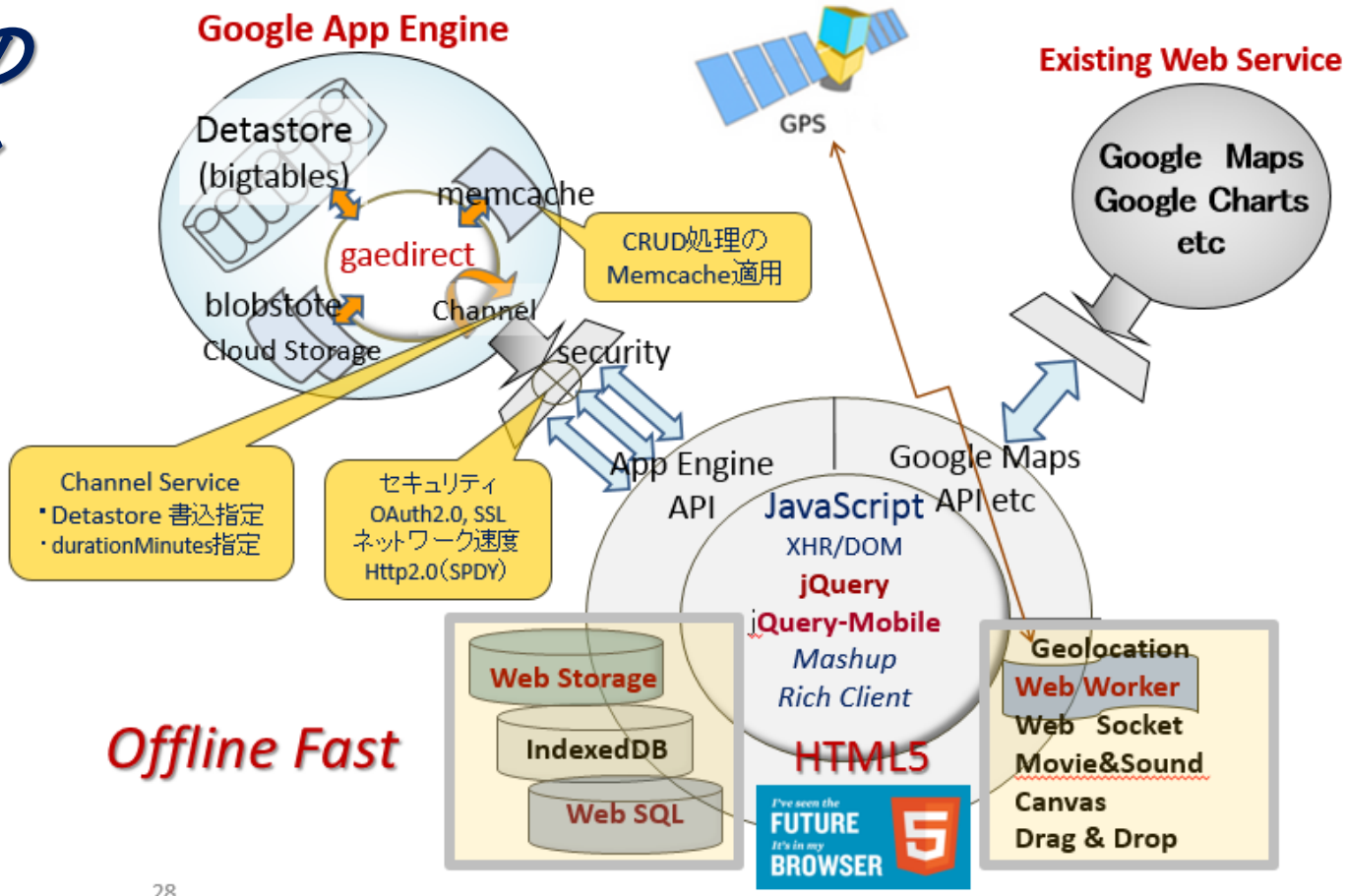


PaaSクラウドの開発生産性を飛躍的に向上させるBaaSツール：gaedirect

gaedirectを使用すればクラウドサーバ側のコード記述を全く行う事なく、ブラウザでのjQueryを使用したJavaScriptのみの記述によってBigTableのCRUD(登録・参照・更新・削除)処理および条件参照を行うことができます。またその他にも、Channelサービスを使用したサーバプッシュ処理や、画像・動画のアップロードなどもWebブラウザ側のコード記述のみでできるようになっています。

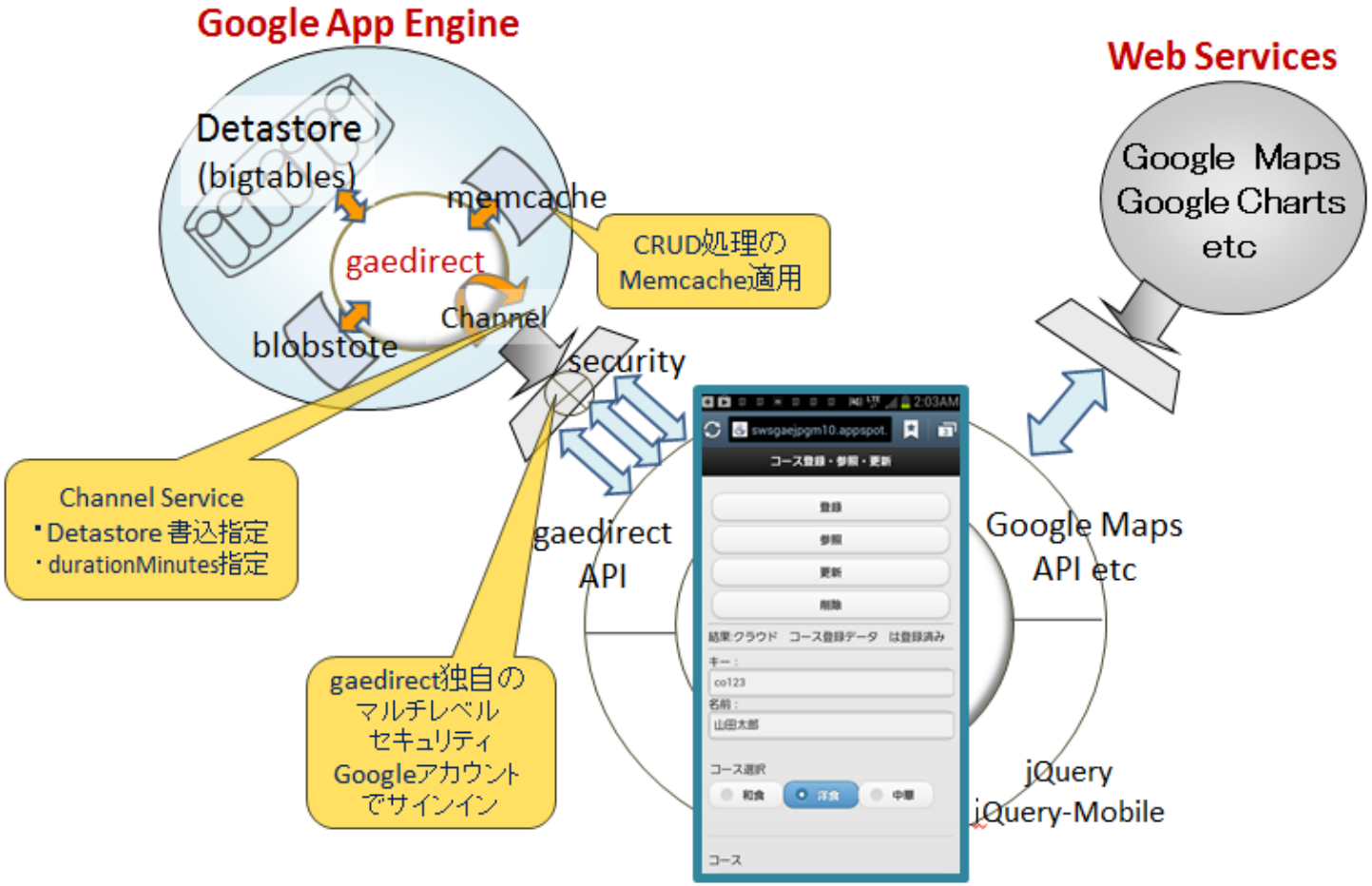
また、Webクライアント側でHTML5のオフラインストレージ機能等を組み合わせればデータストアアクセス等で格段のパフォーマンスを得ることができます。

Cloud Fastの BaaSツール gaedirect



Mobile Fast　BaaSからMBaaSへ

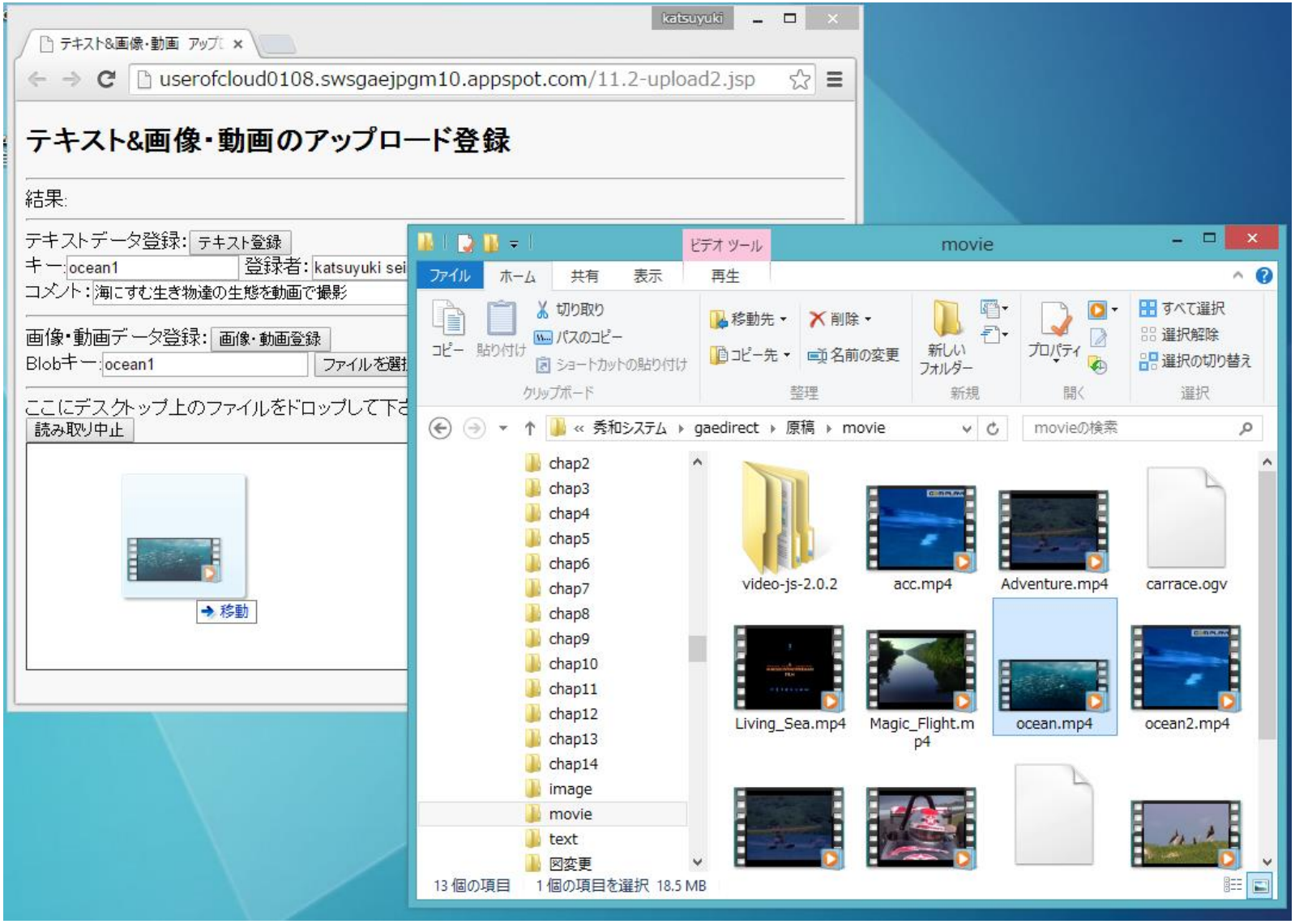
GaedirectでクラウドのDatastore操作追加



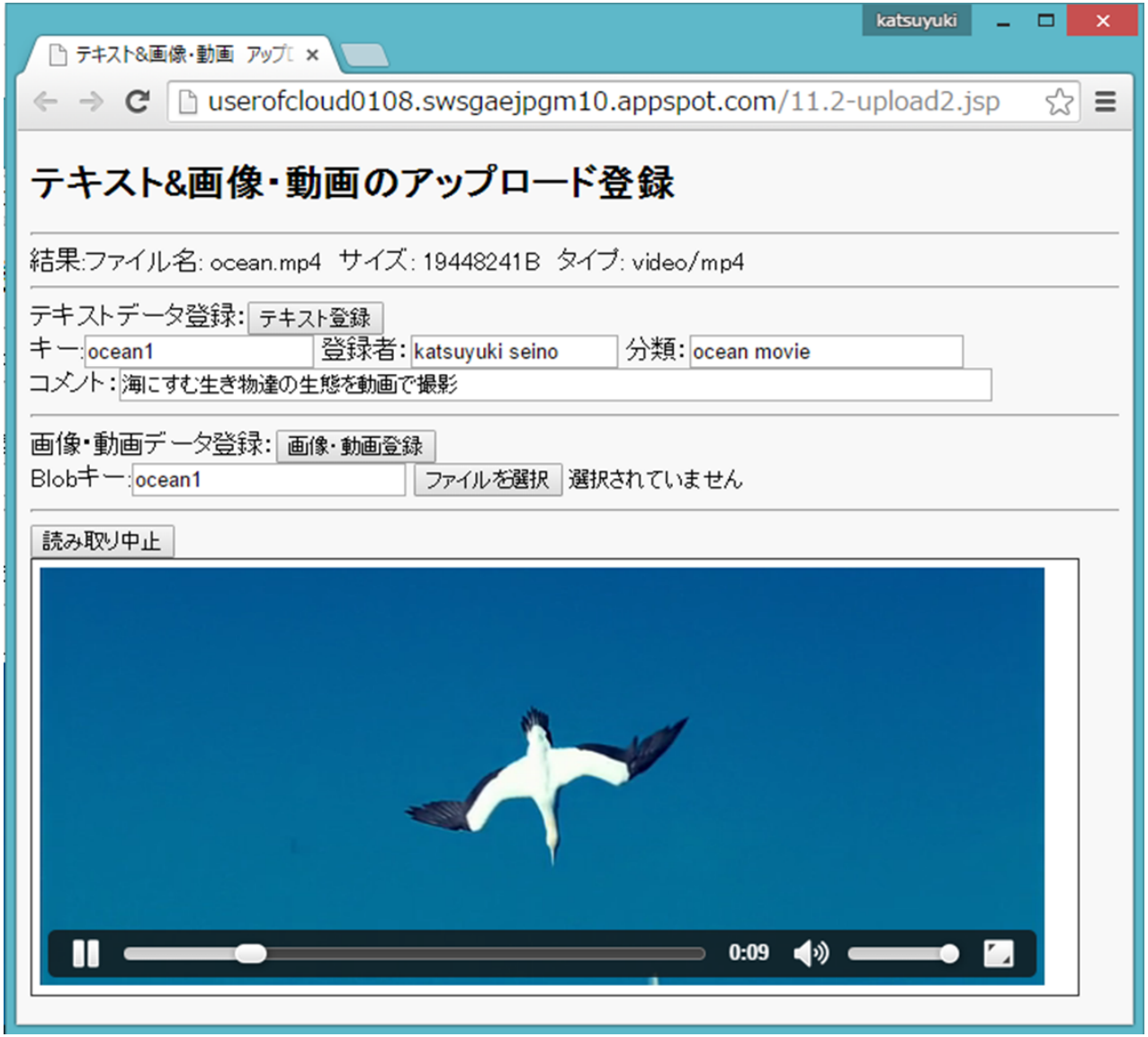
http://superstring-gaedirect32.bizcloudsql.appspot.com/14.1-MultifunctionChat3F.htm

gaedirectでCRUD処理

gaedirectを使用した動画のアップロード



gaedirectを使用したアップロード動画の放映確認

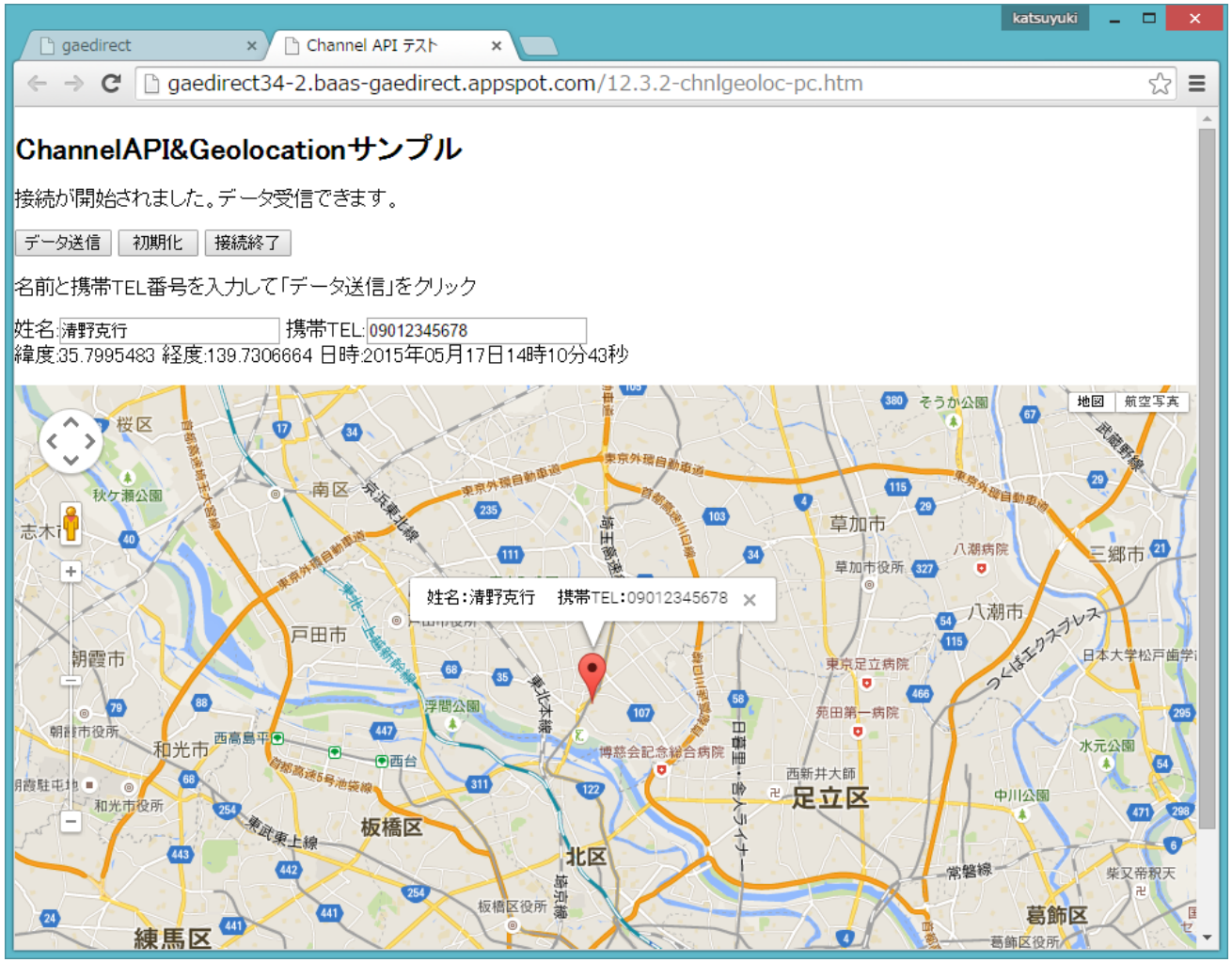


gaedirect

geolocationとChannelAPIで作る緊急連絡用サーバプッシュ表示

gaedirect

スマホの「データ送信」ボタンクリックで、プッシュ送信でPC画面に地図と位置情報のマーカ表示



gaedirectで作る Twitter+（Multifunction Chat）

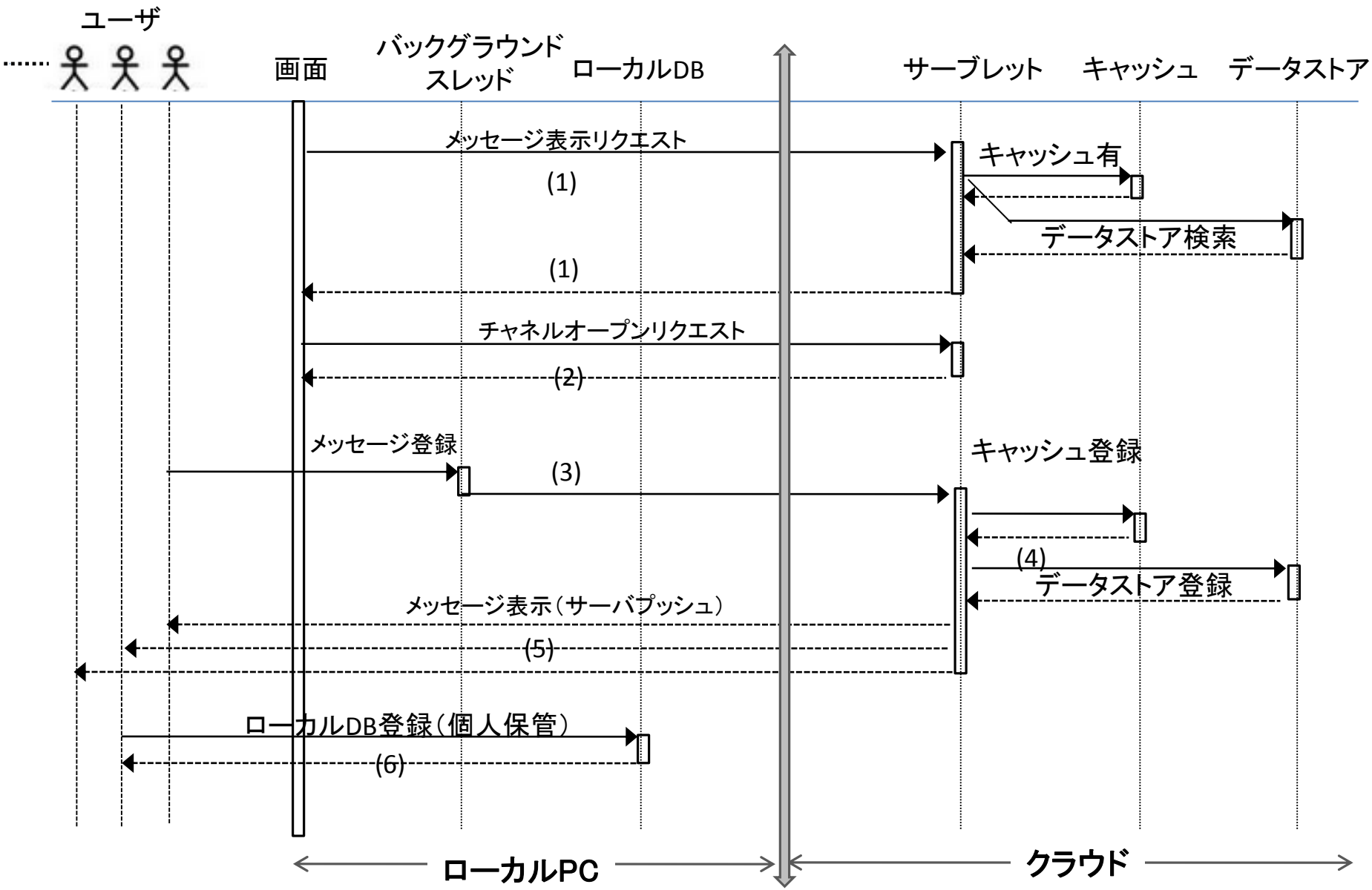
http://superstring-gaedirect32.bizcloudsql.appspot.com/14.1-MultifunctionChat3F.htm

4:38PM
ステータス：
接続が開始されました。データ受信できます。
App Engineデータ参照
ローカルPCデータ参照
ローカルPCデータ削除
Google Maps参照
0 えええええ sssss@gmail.com
書込日時2013:09:18:16:37:32
緯度35.7965962 経度139.7119471
いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい
1 ああああ aaaaaaaa@ggg
書込日時2013:09:18:16:35:03
緯度35.7965907 経度139.7119876
えええええええええええええおおおおおおおおおいいいい
多機能チャット

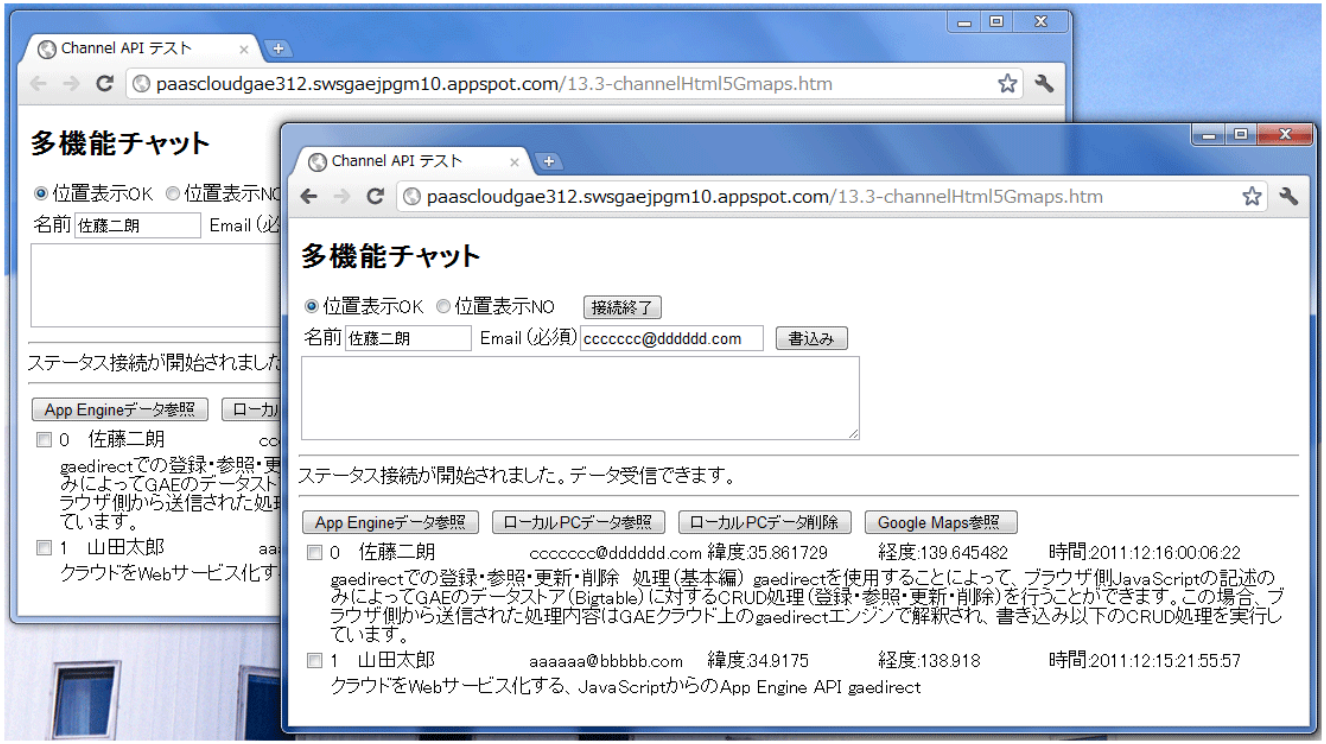
4:44PM

gaedirect

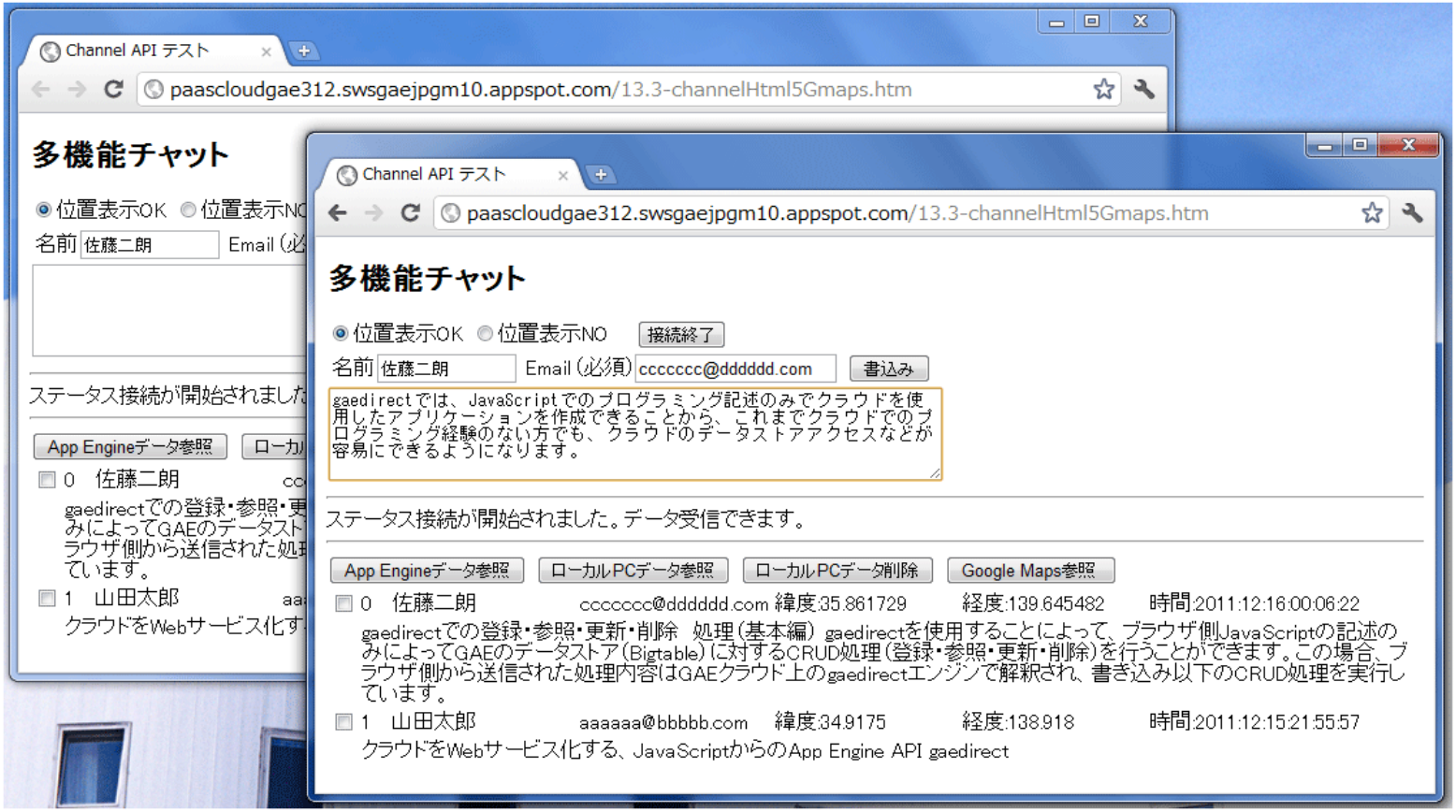
クラウド（App Engine）＋ HTML5でクライアント/サーバ協調オペレーション



gaedirect
多機能チャット初期画面表示

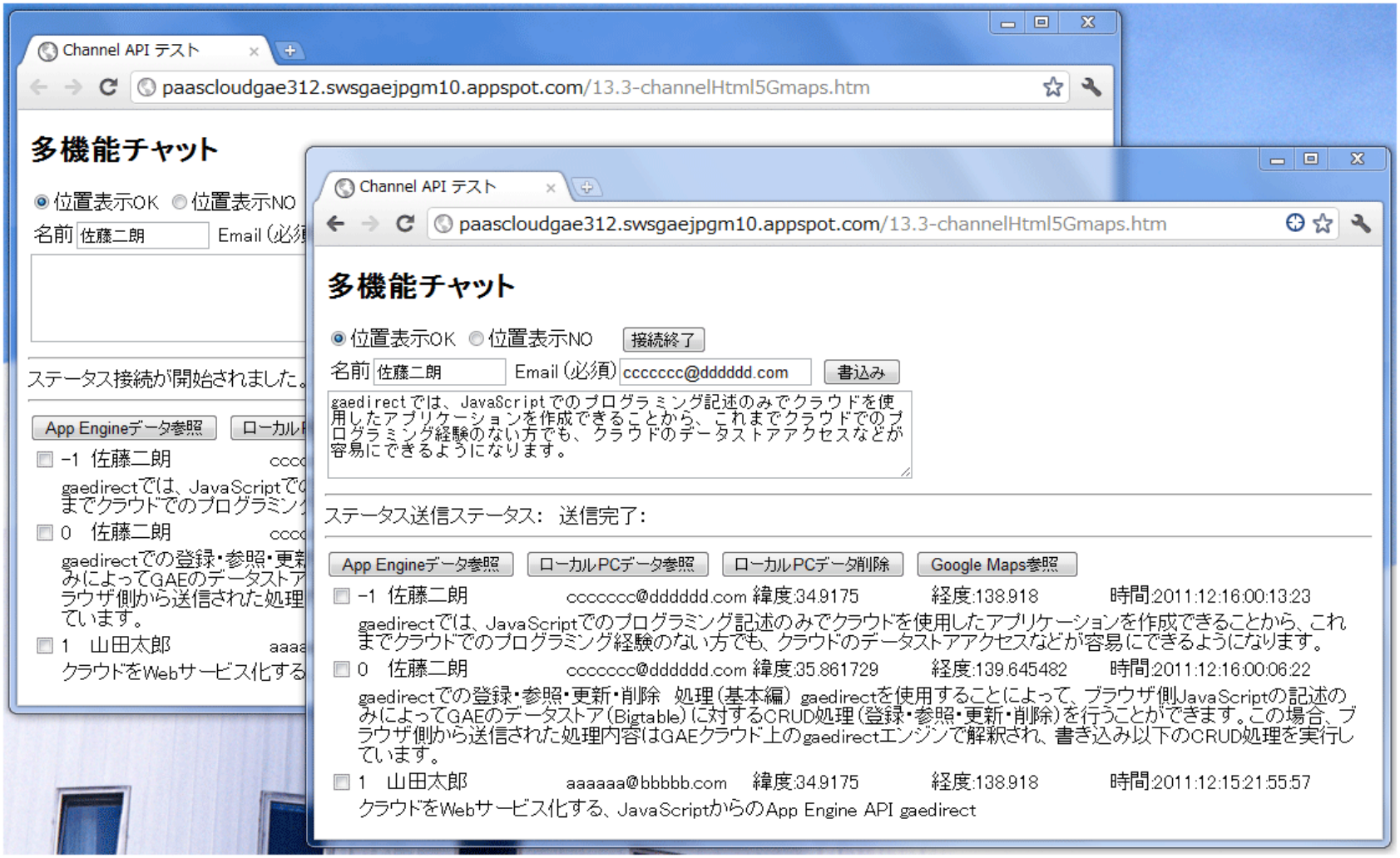


gaedirect
多機能チャット画面にコメント書き込み
チャット参加者が位置表示を行いたくない場合



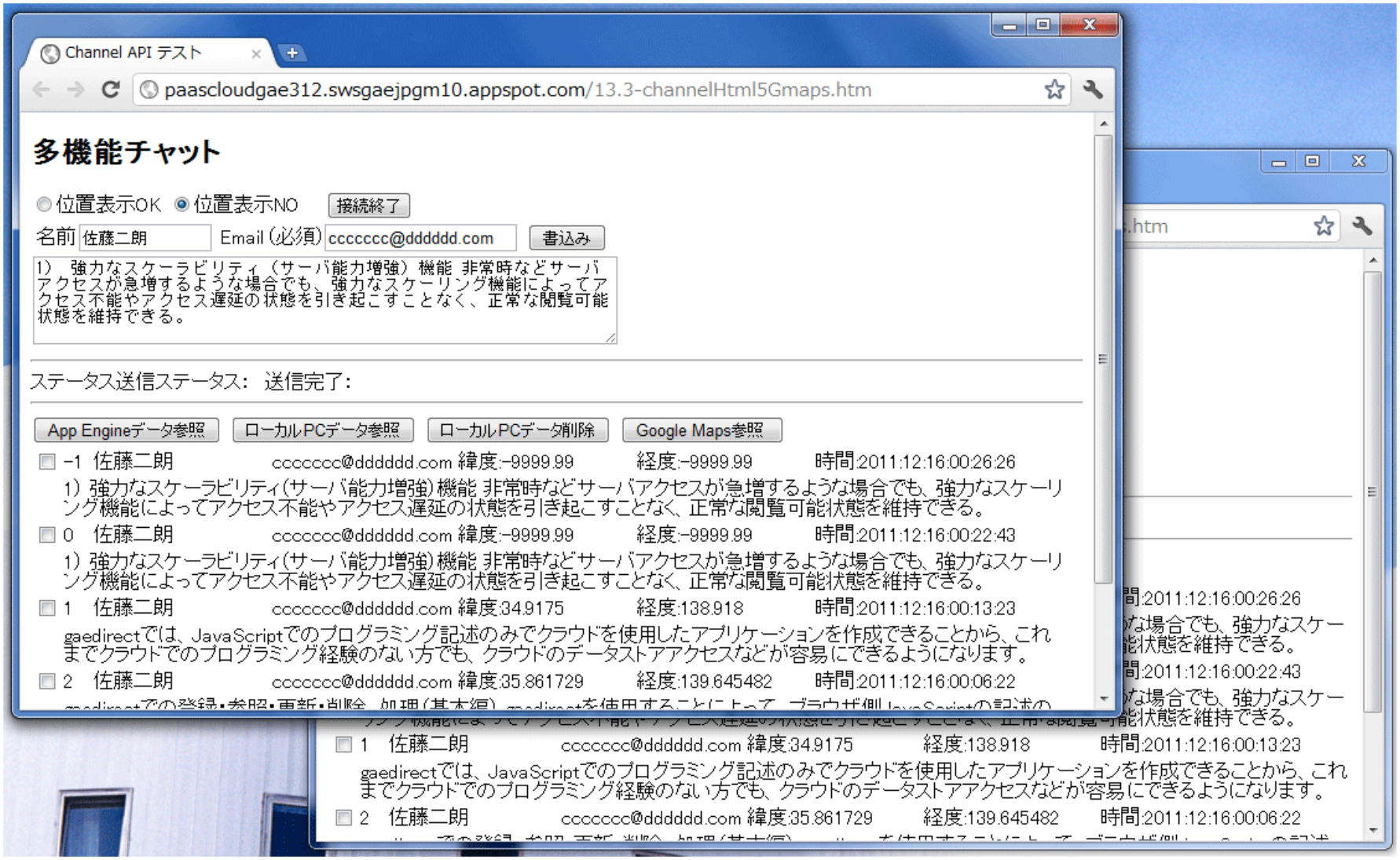
gaedirect

Channel APIのサーバプッシュによる書き込みメッセージ表示



gaedirect

チャット参加者が位置表示を行いたくない場合



gaedirect

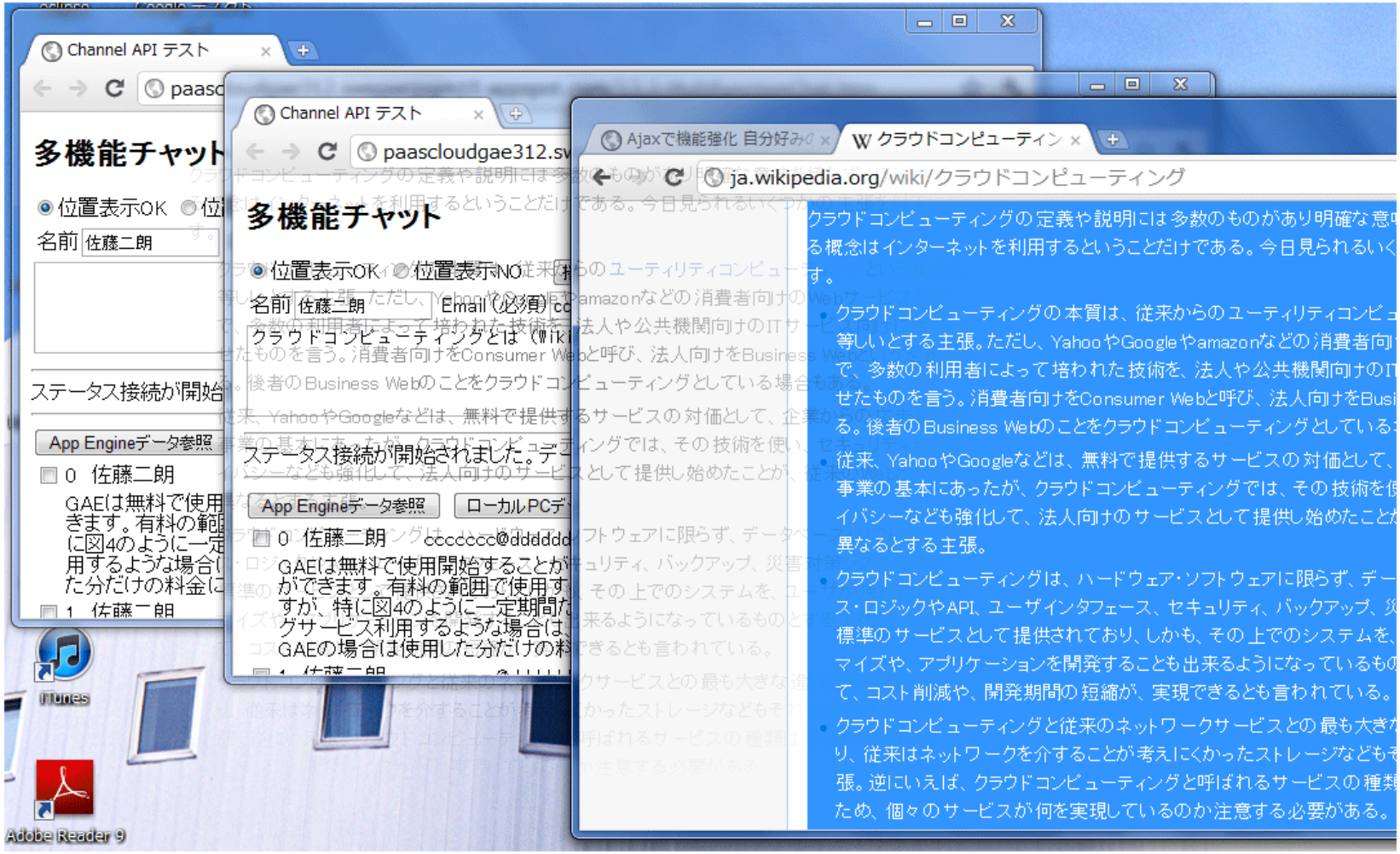
ドラッグ＆ドロップでのメッセージ書き込み

gaedirect

キー入力テキストの後に表示されるドロップテキスト

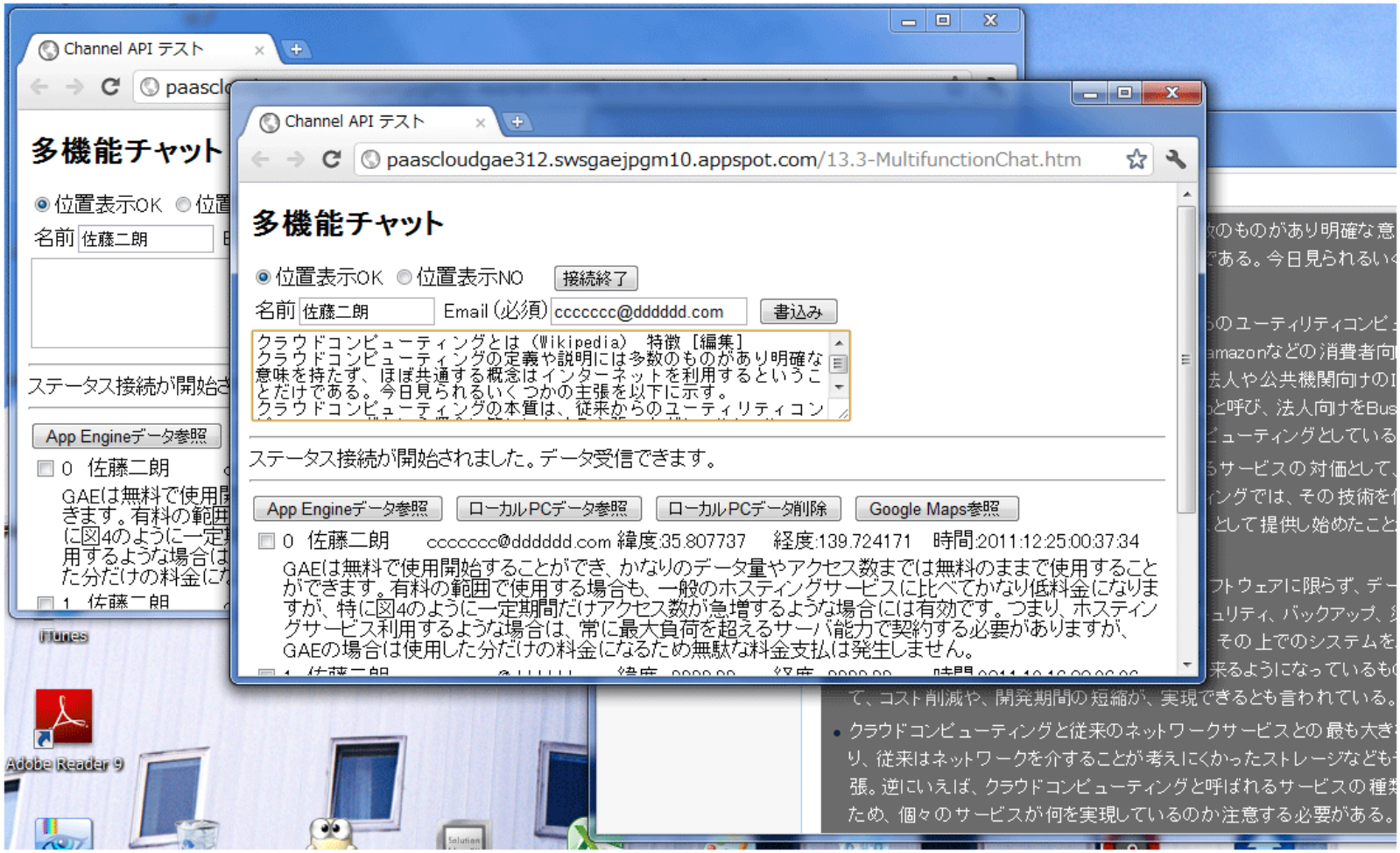
gaedirect

長文テキスト（Wikipediaのクラウドコンピューティング）をドラッグ



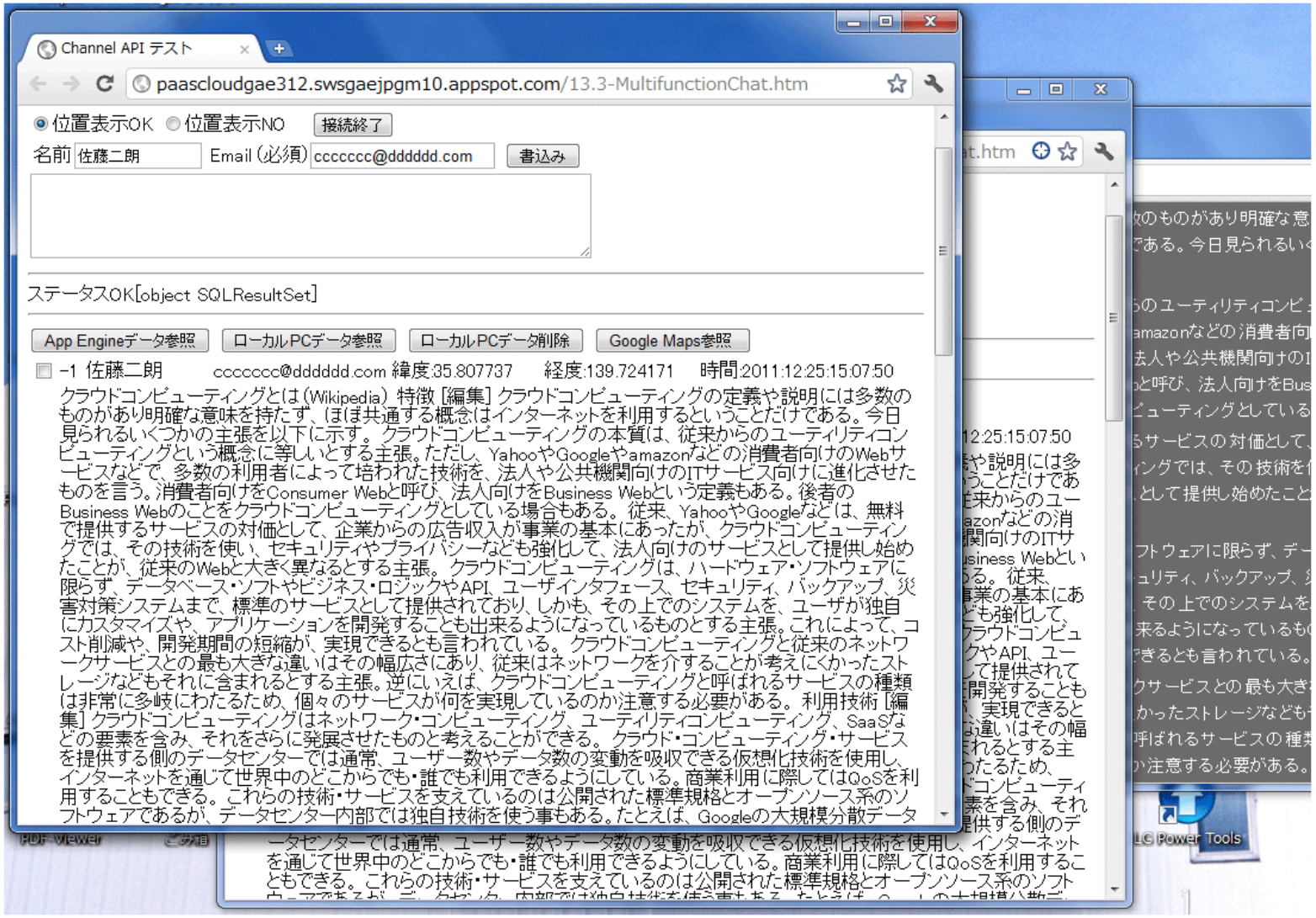
gaedirect

長文テキスト（4000文字超をドロップ）



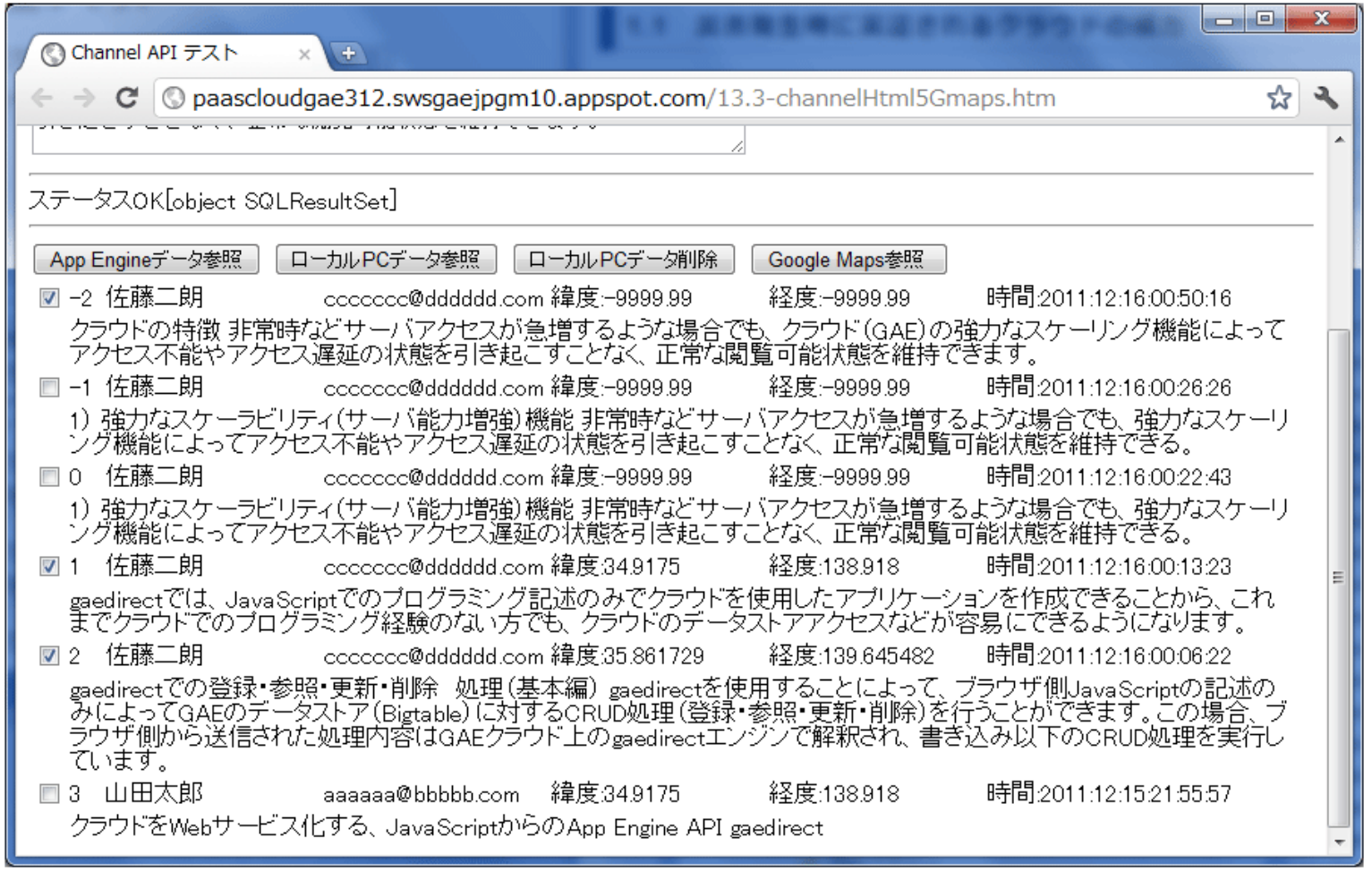
gaedirect

長文テキストのChannel送信とサーバプッシュ表示



gaedirect

ローカルストレージへのメッセージ書き込み



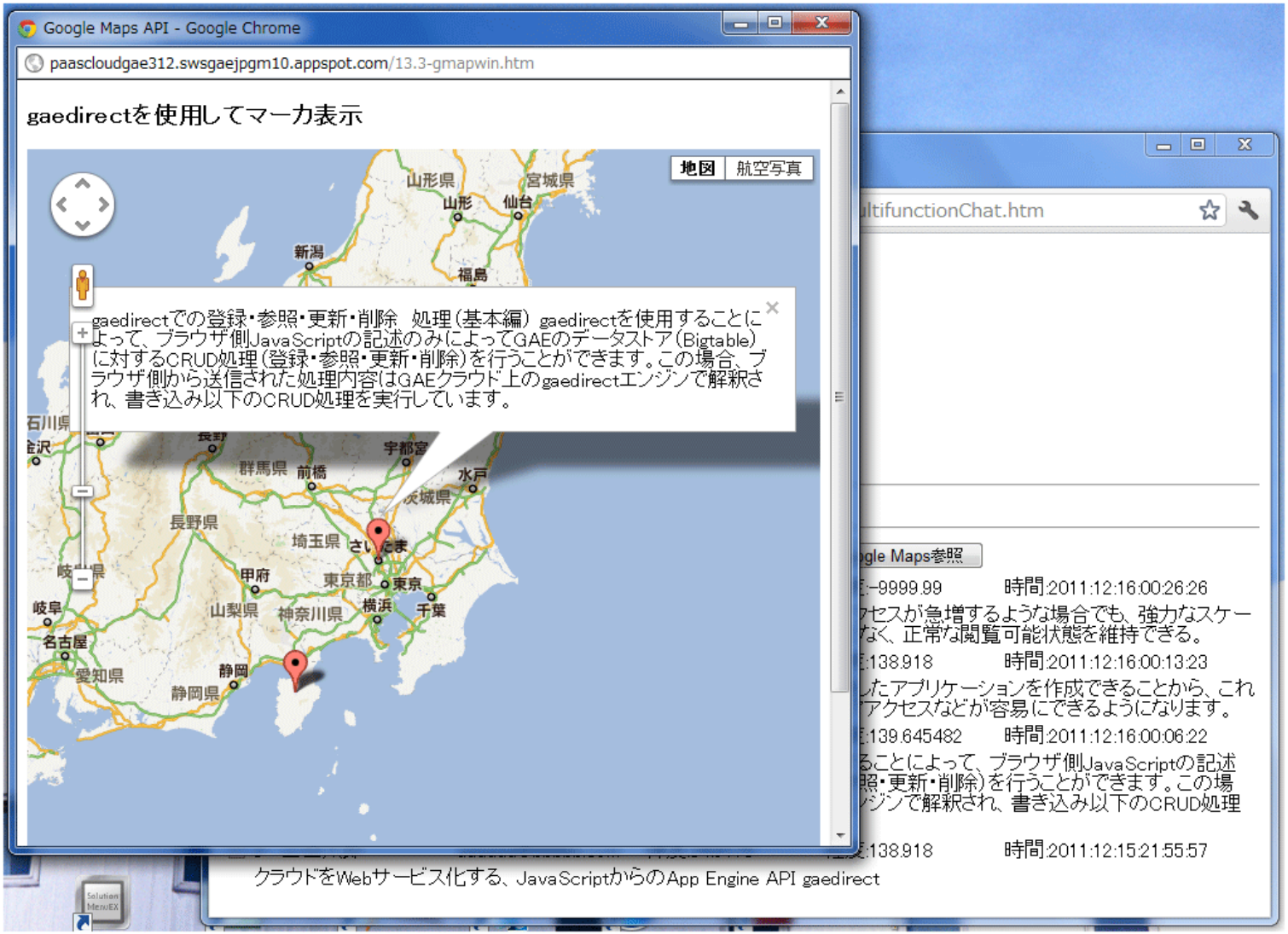
gaedirect
ローカルストレージの内容を確認

gaedirect

チャット参加者のGoogle Maps表示

gaedirect

マーカクリックで最後の書き込み内容を吹き出しで表示



Offline First

今のPCやスマホ等は、CPU処理能力やメモリ容量の増加は目を見張るものがある。これからは、クラウドを設計する場合もWebクライアントまで含めて形での処理を前提にデザインすべきと考えられる。

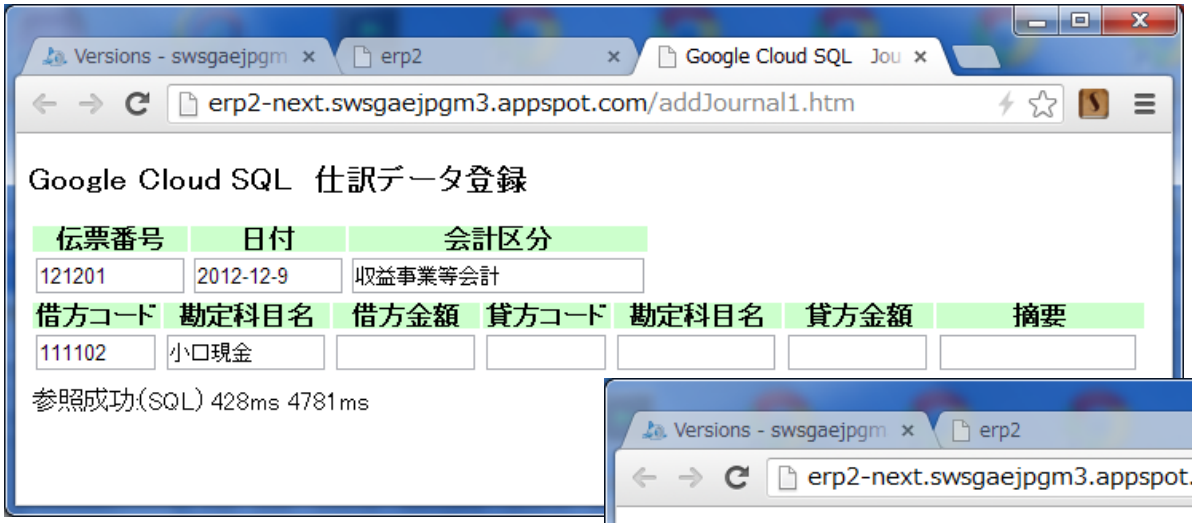
http://journal.bizcloudsql.appspot.com/addJournal1.htm

Cloud SQLとHTML5の連携を掘る

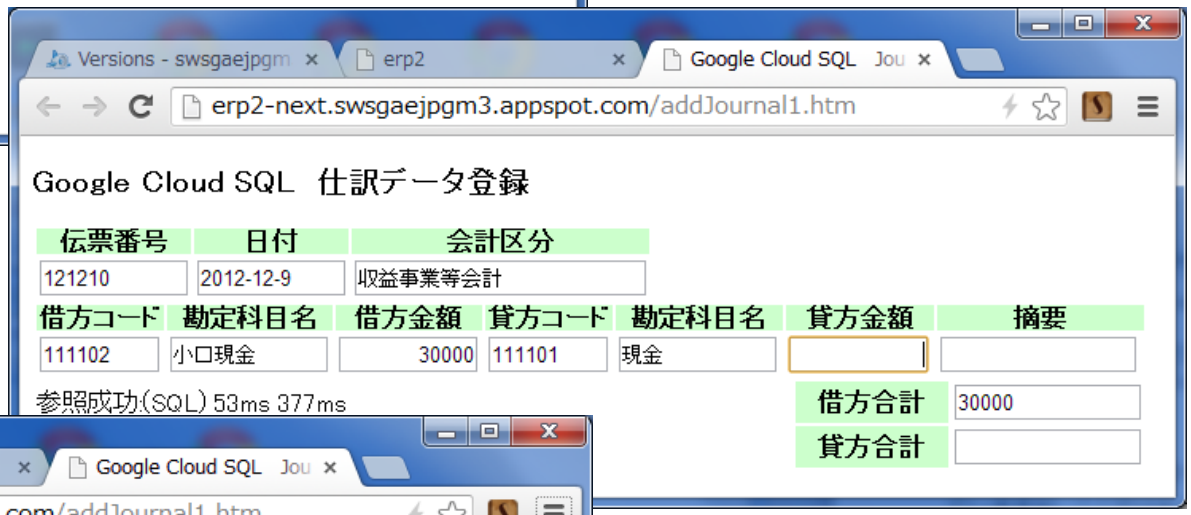
Cloud
- Cloud SQL
- Memcache
- Push Task
- Gson

Client
- Web Storage(HTML5)
- Web SQL Database(Indexed Database)
- Web Workers(HTML5)
- jQuery
- Ajax（Raw code）

仕訳データ入力の処理手順



借方コード入力と
勘定科目名表示



貸方コード入力と
勘定科目名表示

仕訳データの入力完了と
データベース登録

Dig1　Cloud First

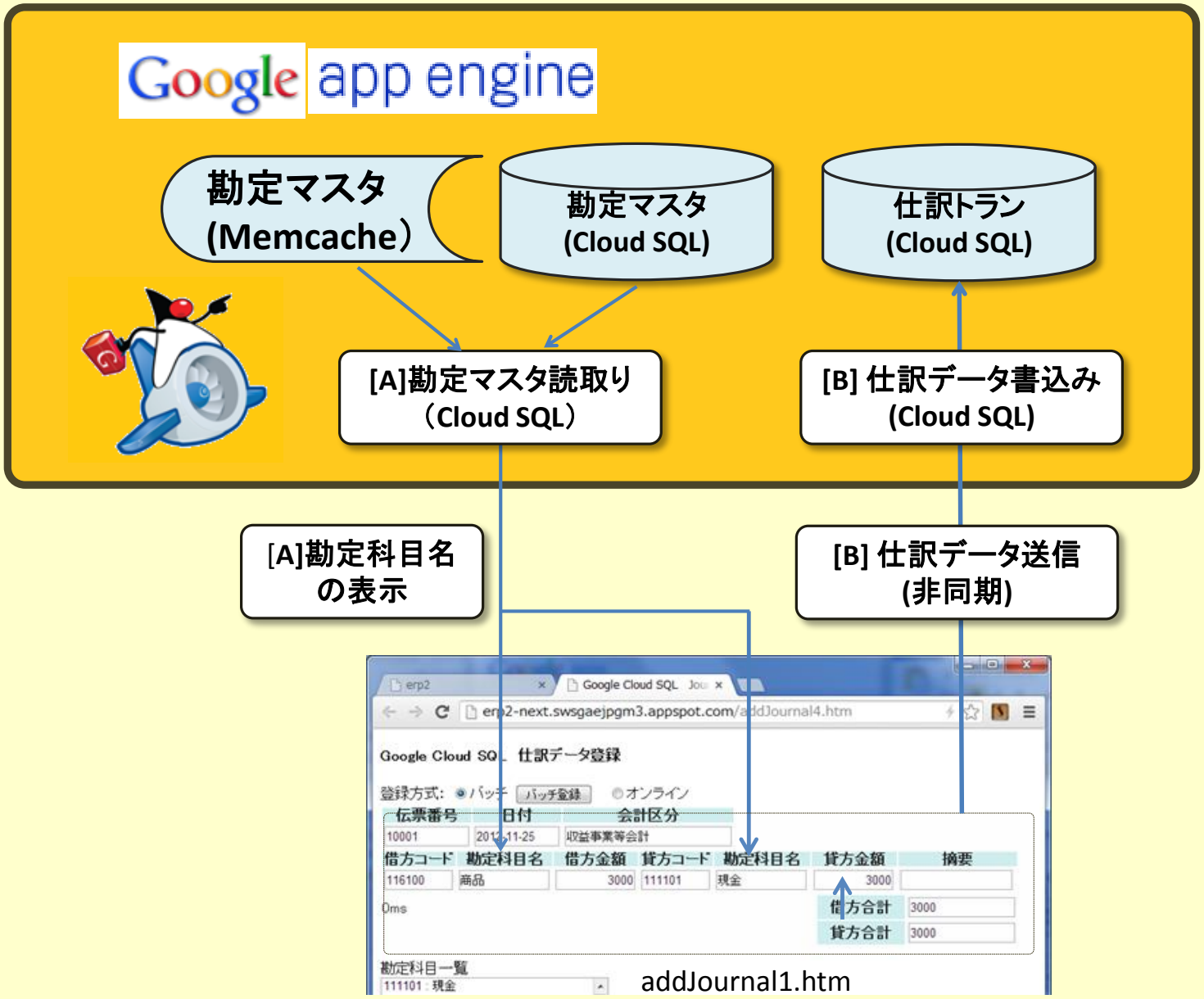
[レスポンス速度の計測]

Cloud SQL アクセス　vs　Memcache アクセス

[結果]

1. App Engineサーバ（Java Beans）で計測した
   アクセス速度では、Memcacheが圧倒的に速いが
2. Webクライアントで計測したレスポンスタイムでは
   それほどの差はなくなっている。

Dig1 Cloud SQL , Memcacheを使用して仕訳データ入力



（1）ストレージアクセス速度

App EngineのJava Beansから下記アクセスでデータ取得に要する時間

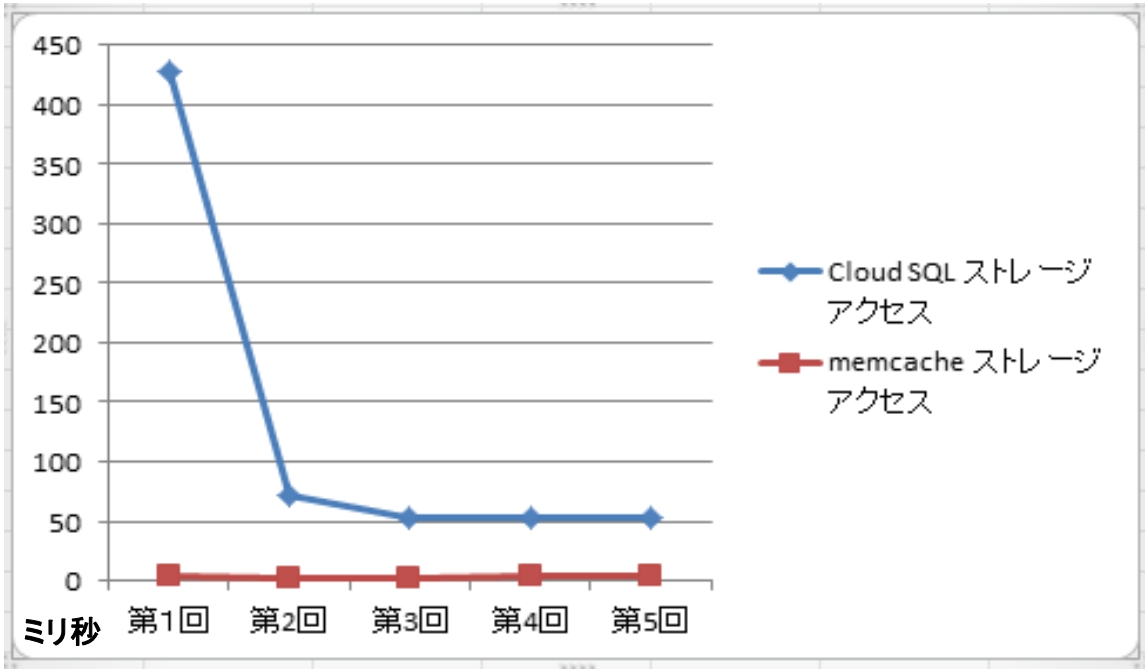
・MySQLに書き込まれた勘定マスタをCloud SQLでアクセス

・Memcacheに書き込まれた勘定マスタをMemcache APIでアクセス

=>Memcacheアクセスが圧倒的に速い！

		第1回	第2回	第3回	第4回	第5回
Cloud SQL	ストレージアクセス	428	72	53	53	52
memcache	ストレージアクセス	3	2	2	3	3

単位ミリ秒



（2）クライアントレスポンス速度

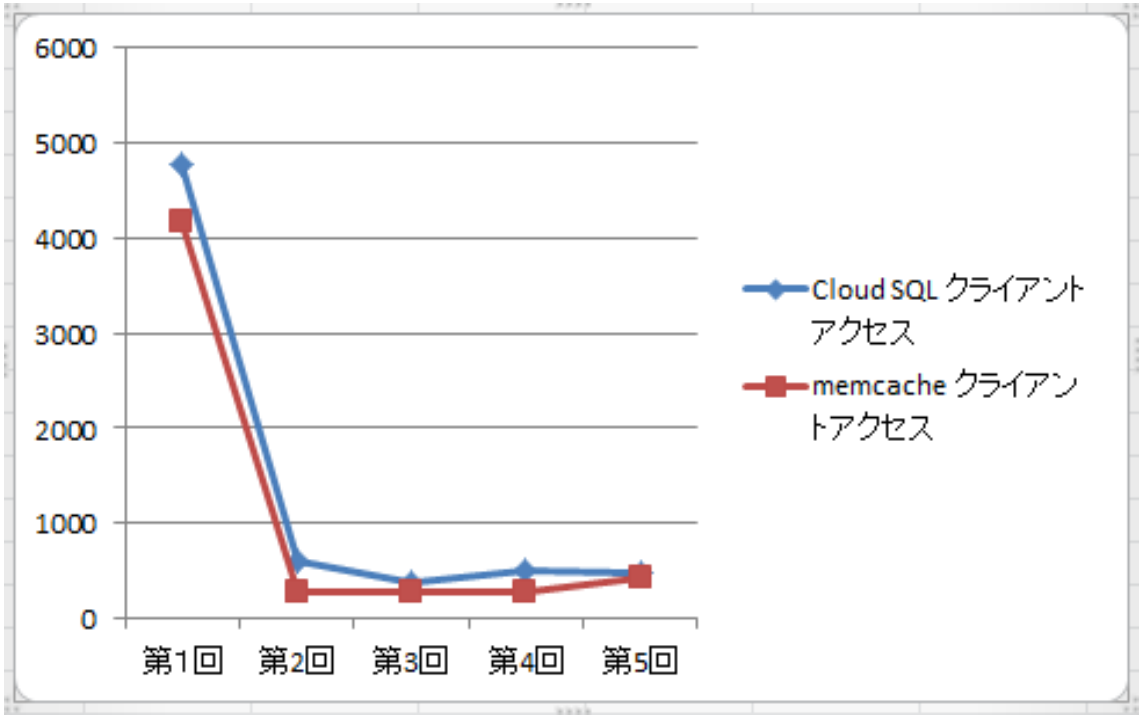
WebクライアントのJavaScriptから下記アクセスでデータ取得に要する時間

- MySQLに書き込まれた勘定マスタをCloud SQLでアクセス
- Memcacheに書き込まれた勘定マスタをMemcache APIでアクセス

⇒ Memcacheを使用してもクライアントレスポンスでは差は縮まる

		第１回	第2回	第3回	第4回	第5回
Cloud SQL	クライアントアクセス	4781	594	385	507	482
memcache	クライアントアクセス	4182	281	284	275	427

単位ミリ秒



Dig2　Cloud First + Offline Fast

[レスポンス速度の計測]

Cloud SQL 、Memcache アクセス

vs

Web Storage（Web Client）アクセス

[結果]

1. App Engineサーバアクセスでは最速のMemcacheアクセスより、Web Storageアクセスの方が圧倒的に速い。

2. Web Storageアクセスではアクセス速度が1ミリ秒以下の場合も多く、人間の感覚からは全くの同時になる。

Dig2 Cloud SQLとWeb Storageによる勘定名超高速表示

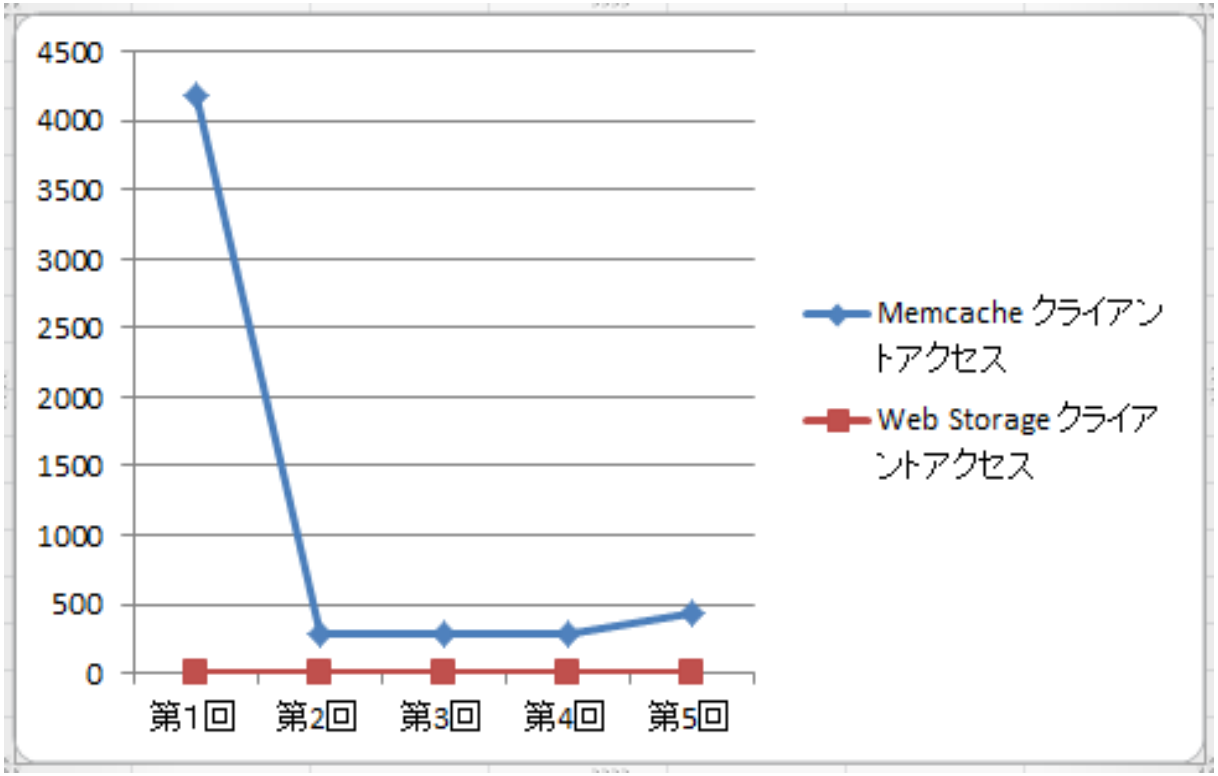
[パフォーマンス測定]

Memcacheと Web StorageのWebクライアントレスポンス速度

=> Web Storage使用で圧倒的な高速レスポンス

		第1回	第2回	第3回	第4回	第5回
Memcache	クライアントアクセス	4182	281	284	275	427
Web Storage	クライアントアクセス	1	1	0	1	0

単位ミリ秒



Examine Network latency

距離

・日本　サンフランシスコ　距離　　5131　mi　=　8257.3 km

・海底ケーブル

http://trendy.nikkeibp.co.jp/article/column/20091109/1030118/?rt=nocnt

全長9,600km　　往復　19,200　km

Overhead　　往復　　800　km

――

合計　　20,000　km

速度

光速　　299 792.458 m/s

ガラス媒体中の高速度　−>　屈折率分の1の速度

ガラスの屈折率が1.5として、ガラス内での光速は200,000km/s

所要時間

従って　日本とUS西海岸での所要最短時間は

20,000 [km]/200,000[km/sec]　=　0.1sec　=　**100ms**

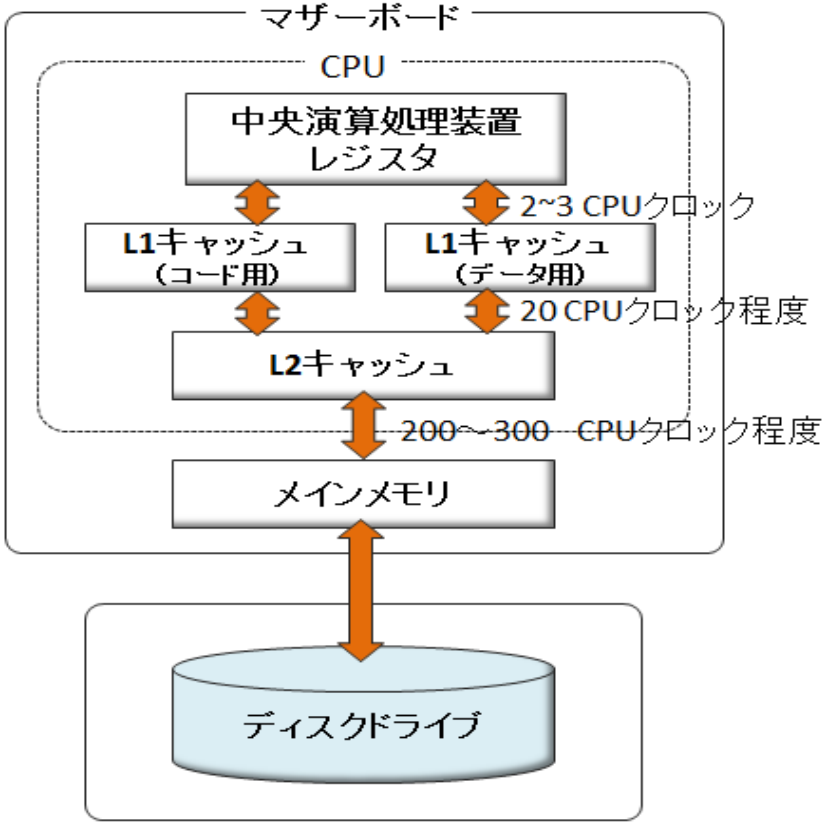
network latency最少でのmemcache Webクライアント最速タイム ≒ 103ms

=>理想環境で現実にはあり得ない

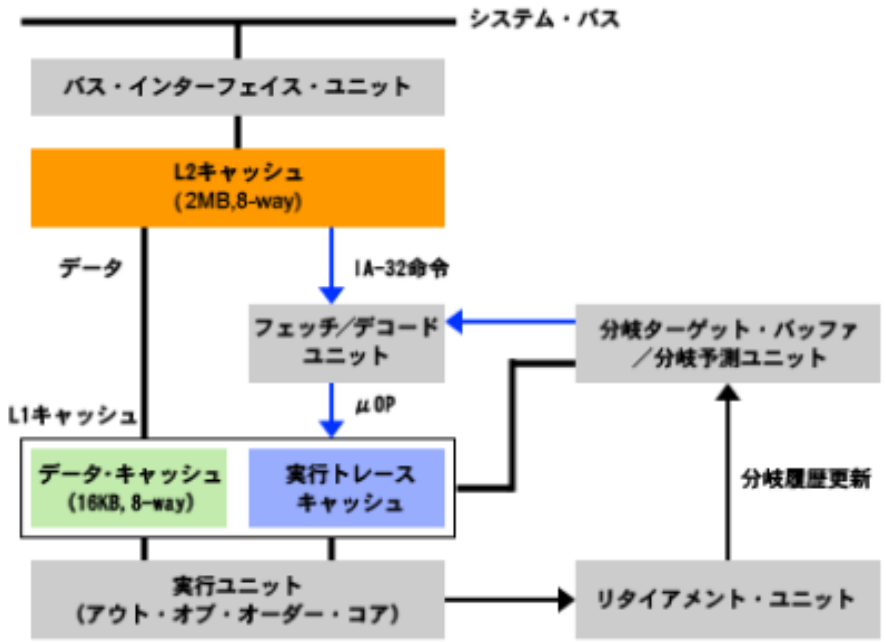
=>Web Storageアクセスでは　**1ms**　またはそれ以下

Examine Local Storage Access

コンピュータのデータアクセス



Intel IA-32 のCPUとL1, L2キャッシュ



Dig3 Cloud First + Offline Fast

Web Storageの利用で
データエントリは実質待ち時間ゼロ！

Web Workersを使用してApp Engineサーバへの
データ送信をバッククラウンドスレッドで行う

Dig3 Cloud SQLとWeb Storage,Web Workersにより
完全待ち時間なしの仕訳入力

Google app engine

勘定マスタ
(Cloud SQL)

[A]勘定マスタ読取り
（Cloud SQL）

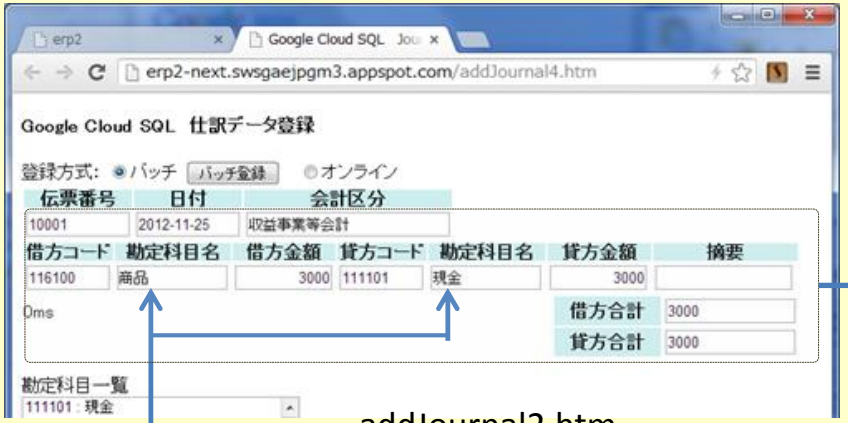
Background sled

勘定マスタ
(Cloud SQL)

[C] 仕訳データ書込み
(Cloud SQL)

HTML

[A]勘定科目の
Web Storage書込み



addJournal2.htm

HTML

[C] 仕訳データ送信
(Web Workers)
バックグラウンドスレッド
で非同期送信

勘定科目マスタ
(Web Storage)

[B]勘定科目名の表示

Dig4　Cloud First + Offline First

Web SQLを使用して入力した仕訳データの登録をローカルストレージに行う。

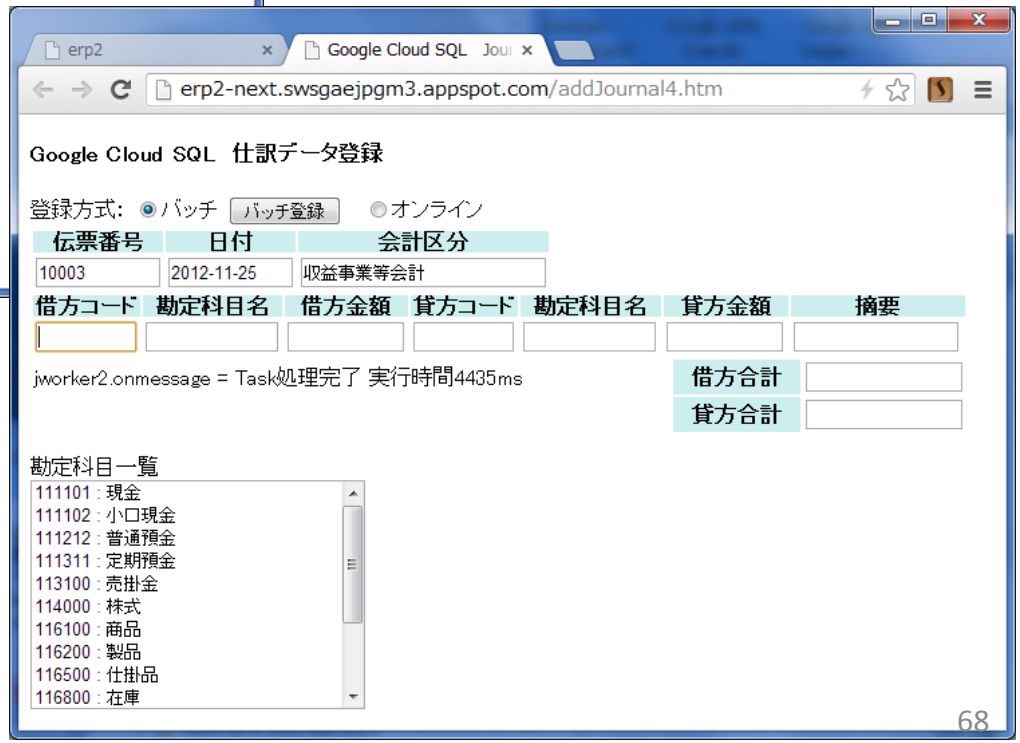
↓

[Offline First]

ネット接続のないオフライン環境でデータエントリか可能

ネット接続なしのオフライン仕訳入力

仕訳データをWeb SQLで
ローカルストレージ登録



Push Taskで
複数仕訳データサーバ送信

Dig4 Web SQL Web Workers Push Taskで
オフライン仕訳入力

Google app engine

Offline First
TSA（Thin Server Architecture）

仕訳トラン
(Cloud SQL)

Task Queue
の処理条件
queue.xml

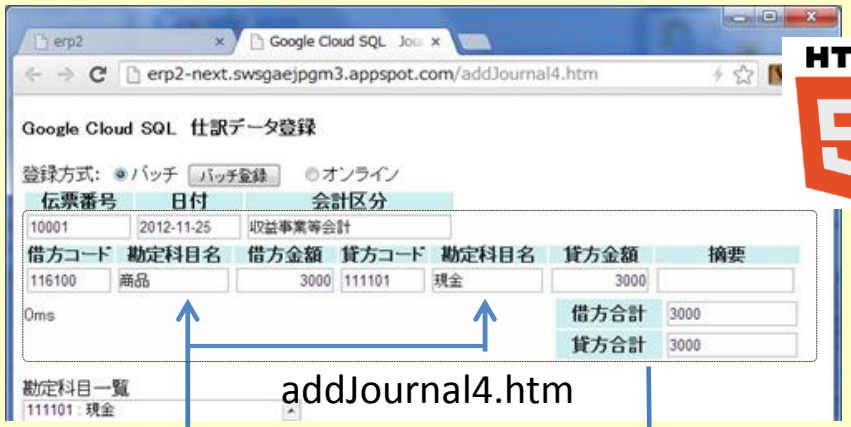
[H]Workerサーブレット
（Task Queues）
acctWorkerServlet.java

Task Queue

[G]Pushサーブレット
Gson
acctTaskJsonServlet.java

勘定マスタ
(Cloud SQL)

HTML



addJournal4.htm

HTML

[E]仕訳データ全件を送信
（Web Workers）
Jworker2.js

[D]仕訳データ全件検索
（Web SQL）

[A]勘定科目のローカルストレージ書込み

勘定科目マスタ
(Web Storage)

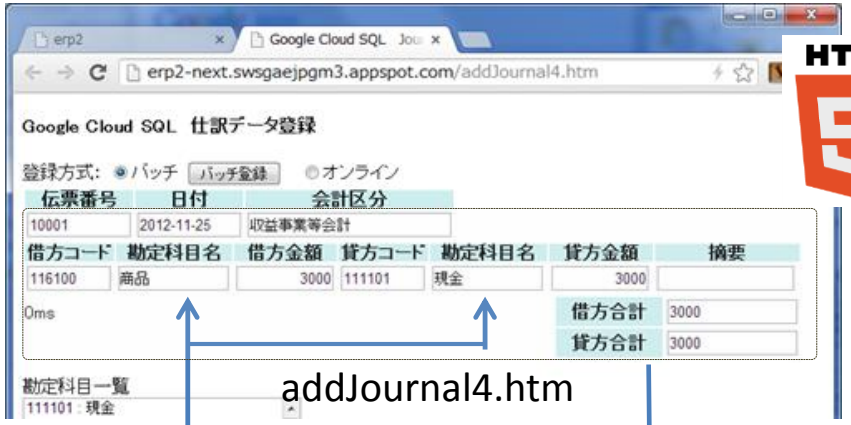
[B]勘定科目名の表示

[C]仕訳データ書込み
（Web SQL）

Σ仕分けデータ
(Web SQL)

Dig4 Web SQL Web Workers Push Taskで

Offline Fast! -> Offline First!



addJournal4.htm

HTML

[A]勘定科目のローカルストレージ書込み

勘定科目マスタ
(Web Storage)

[B]勘定科目名の表示

[C]仕訳データ書込み
（Web SQL）

Σ仕分けデータ
(Web SQL)

HTML

[D]仕訳データ全件検索
（Web SQL）

[E]仕訳データ全件を送信
（Web Workers）

Jworker2.js

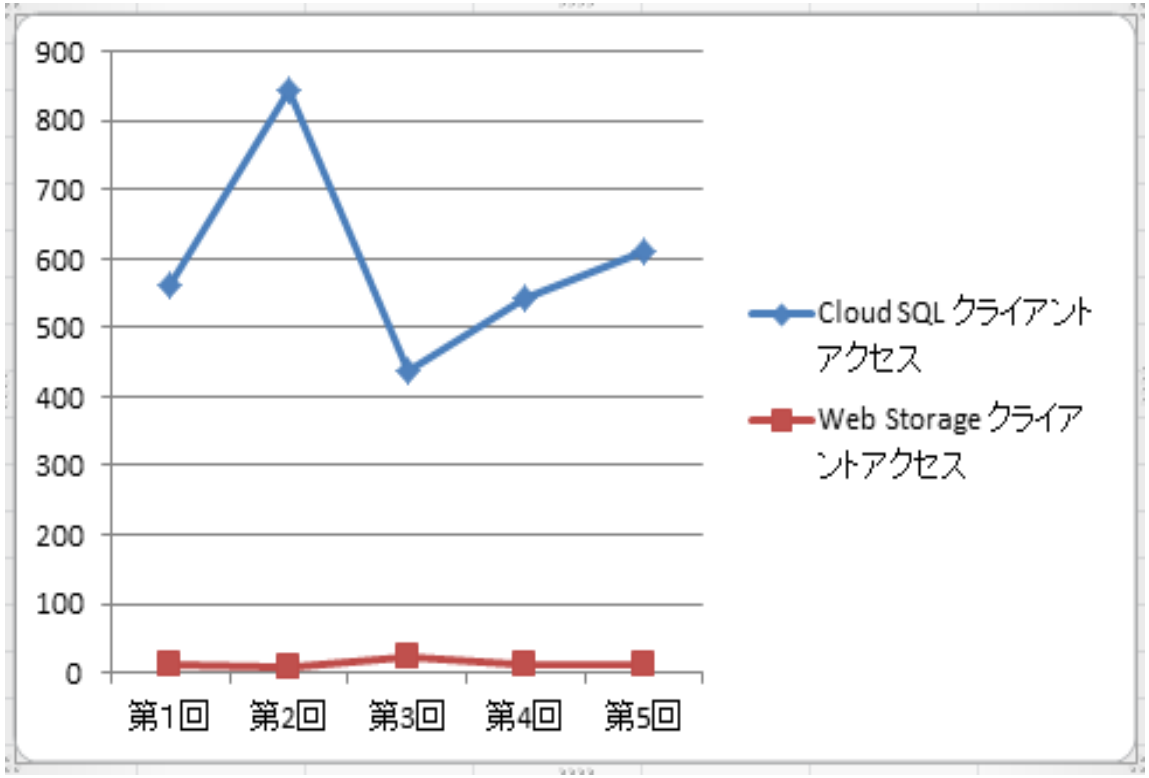
[パフォーマンス測定]

仕訳データの登録　Cloud SQL　vs　Web Storage

=> オフライン入力可能でさらに圧倒的な高速レスポンス

		第1回	第2回	第3回	第4回	第5回
Cloud SQL	クライアントアクセス	560	843	438	542	611
Web Storage	クライアントアクセス	11	9	22	11	10

単位ミリ秒



Dig No2

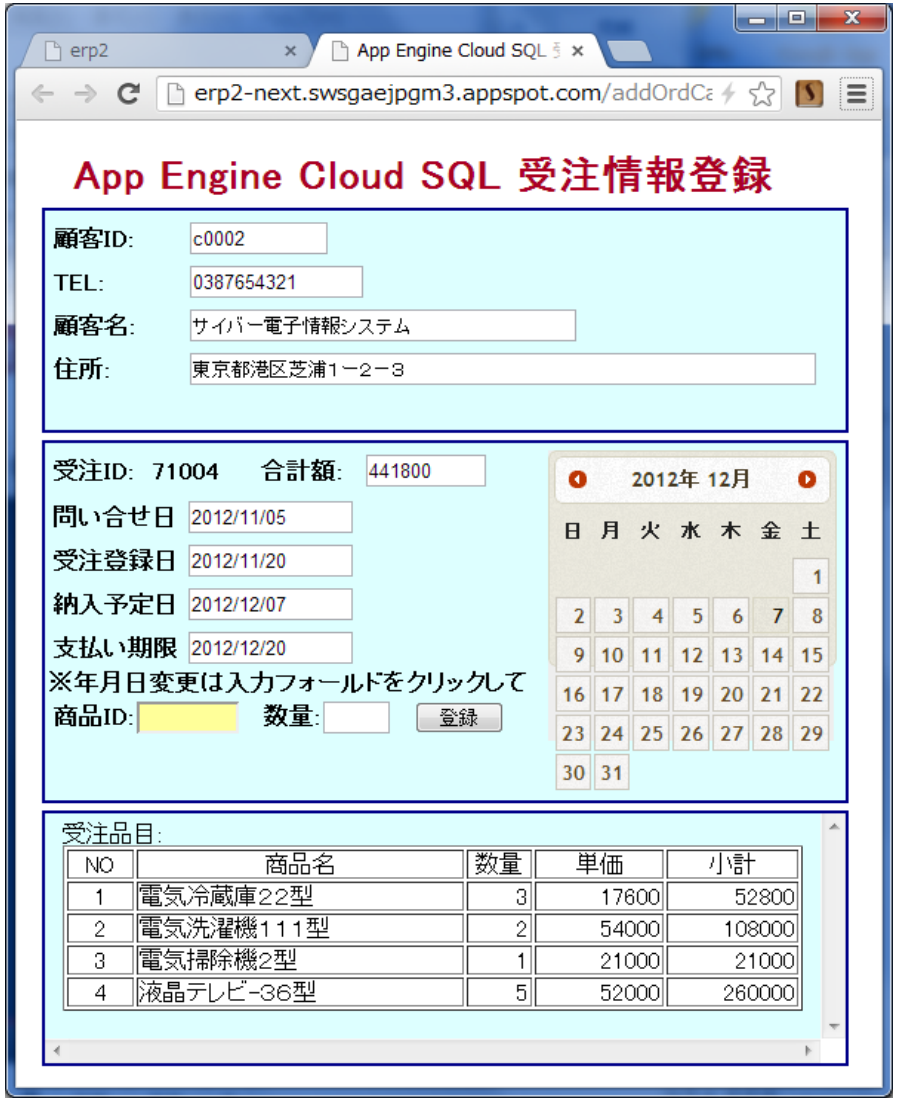
App Engineとクライアントの連携を掘る

Cloud
- Channel service
- Pull Task
- SQL vs NoSQL（merit & demerit）
- SQL to NoSQL conversion

Client
- Web Workers
- Web SQL
- jQuery、jqPlot
- Ajax(Raw code)
- DOM

Dig1 Cloud First + 基幹系UX

App Engineクラウドで受注情報登録



[機能]

1 受注ID自動表示
=>Cloud SQL Auto Increment

2 顧客IDまたは顧客TELから
顧客情報自動表示
=>Web SQL

3 カレンダクリックで日付入力
=>jQuery UI Datepicker

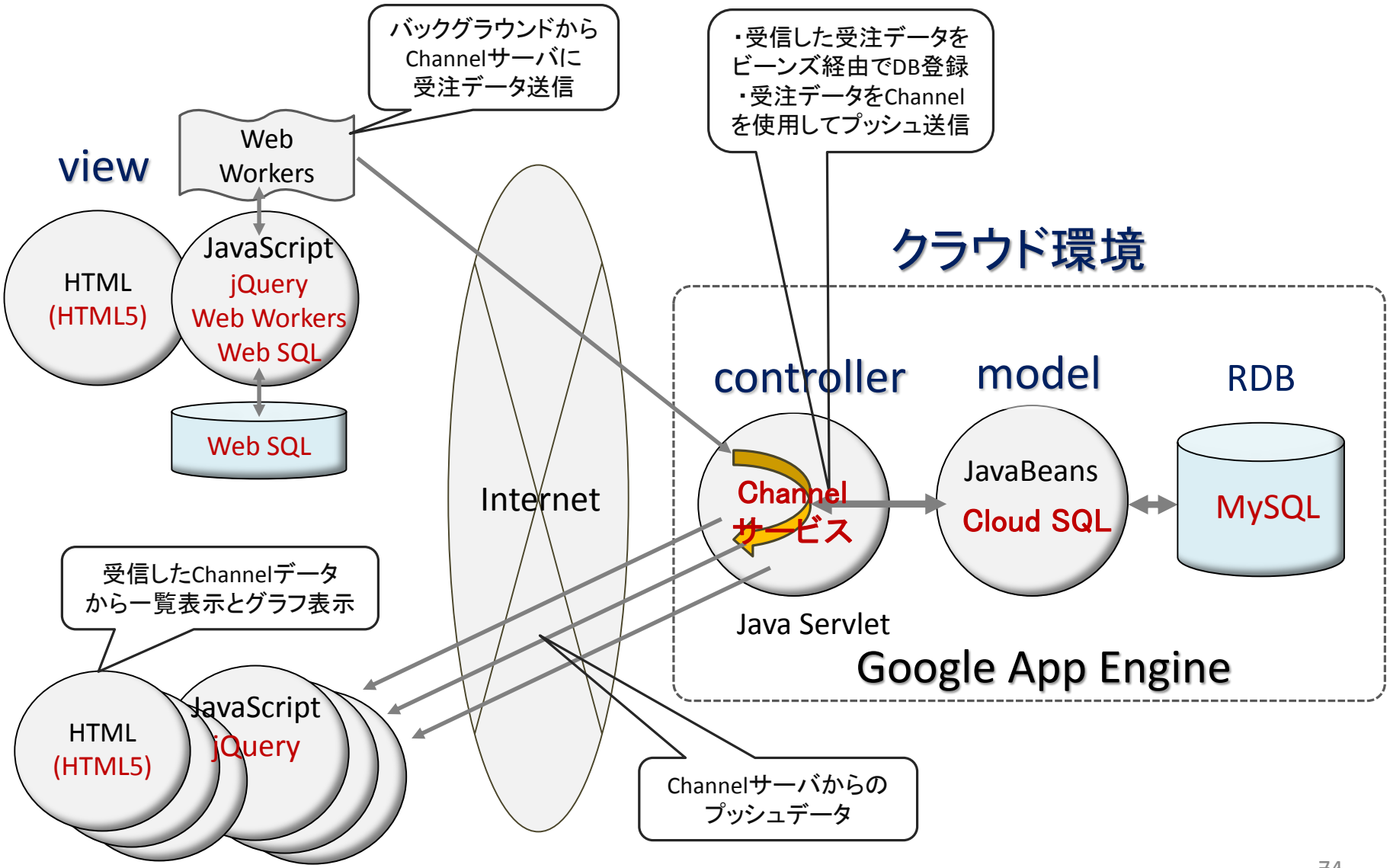
4 商品IDと数量入力で
- 商品名自動表示
- 単価自動表示
- 小計計算と自動表示
- 合計額計算と自動表示

=>Cloud SQL参照&計算 & DOM

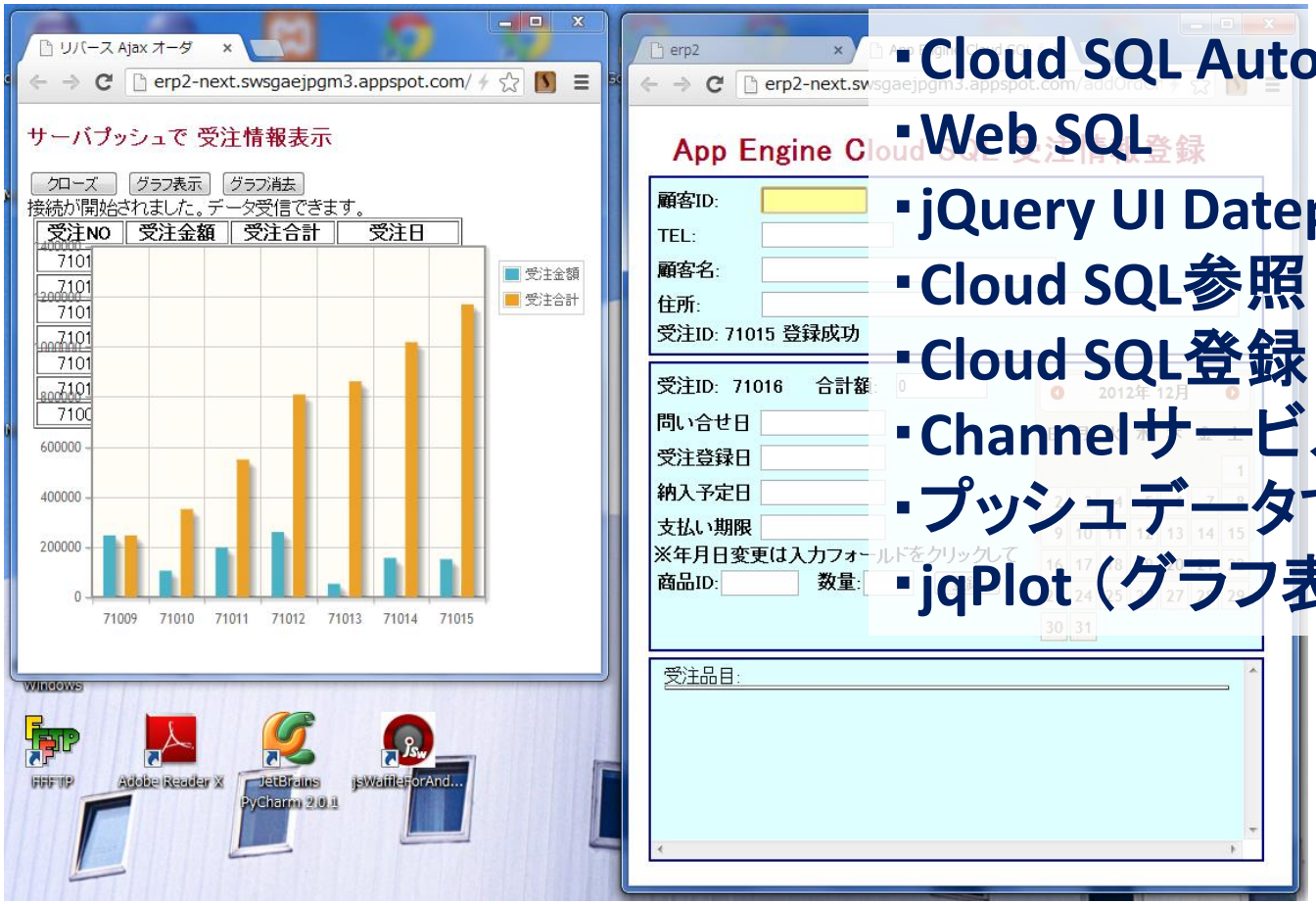
5 ALTキー押下で受注登録
=>Cloud SQL登録

Dig2 Cloud First + UX連携

Channelサービスで、受注サマリをサーバプッシュ表示



App EngineのChannelサービスを使用して、
- 管理者画面に受注サマリをプッシュ表示（DOMスクリプト）
- サマリー覧からグラフ表示（jqPlot）



- Cloud SQL Auto Increment
- Web SQL
- jQuery UI Datepicker
- Cloud SQL参照＆計算 & DOM
- Cloud SQL登録
- Channelサービス
- プッシュデータでDOM表示
- jqPlot（グラフ表示）

Dig3 Cloud First + UX3連携

App Engine

ord_hdr
ord_cal
ord_dtl

Cloud SQL(SQL)

Datastore
(NoSQL)

spaceBean
（LLAPI）

spacemas

channelServlet
Channel
Service

orderBean
（LLAPI）

orderServlet

Cloud SQL(SQL)

cust_mas

specServlet

Channel
プッシュデータ

HTML5
JavaScript
jQuery,jqPlot
DOM

XMLHttpRequest

Web Workers

HTML5
JavaScript
jQuery

cust_mas

Web SQL

HTML5
JavaScript
jQuery

showOrdChnl.htm

addOrdDtl.htm

addOrdDtl.htm

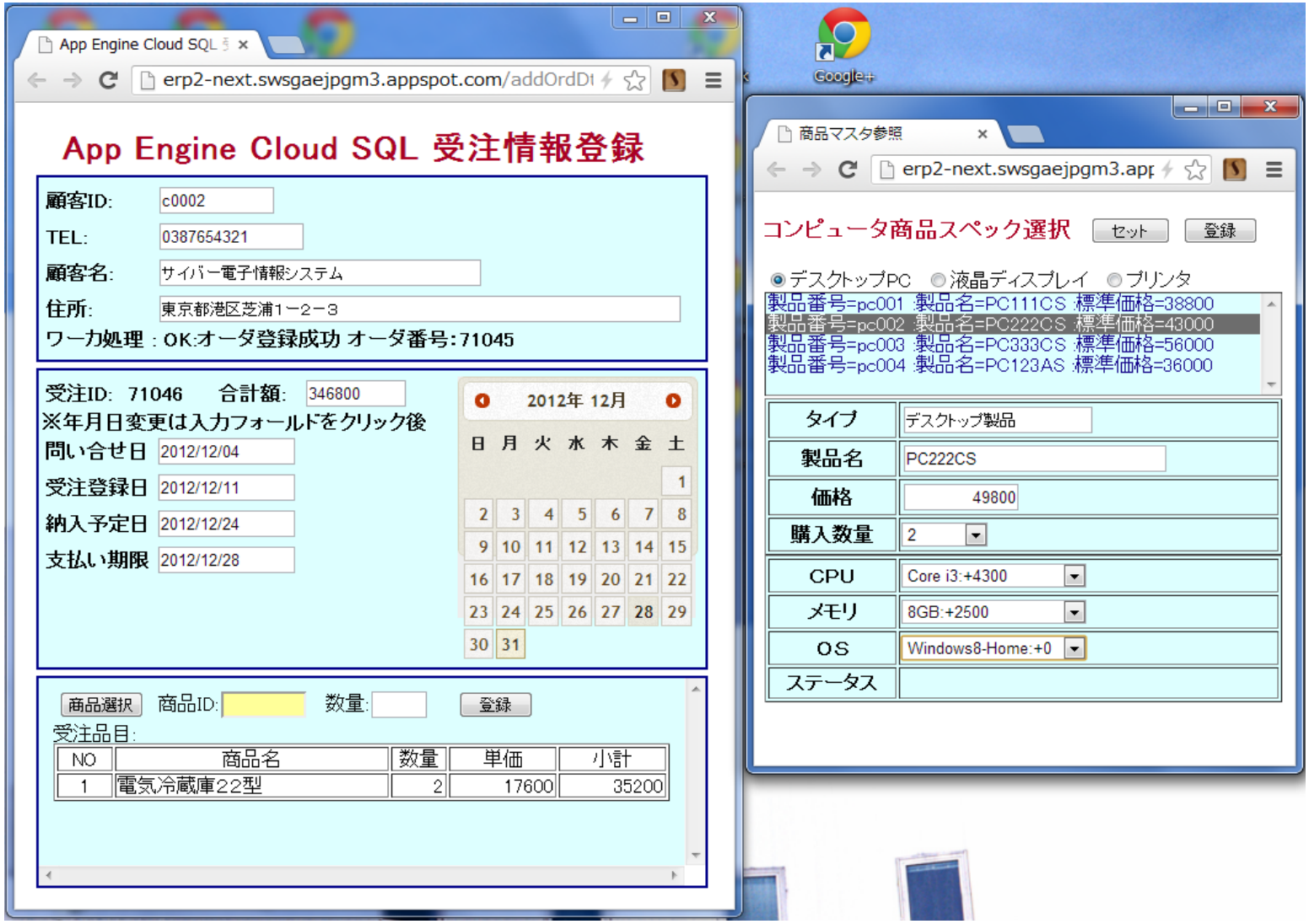
サーバプッシュ

起動

制御

K.Seino

「商品選択」でサブ画面を表示し、サブ画面から商品スペックを指定して
「セット」ボタンをクリックするとメイン画面に商品情報が表示される。



受注品目入力完了後、サブ画面の「登録」ボタンをクリックすれば受注登録が実行され、左の受注情報表示画面に受注サマリが表示される。

- Cloud SQL Auto Increment
- Web SQL
- jQuery UI Datepicker
- Cloud SQL参照＆計算 & DOM
- Cloud SQL登録
- Channelサービス
- プッシュデータでDOM表示
- jqPlot（グラフ表示）
- サブ画面からメイン画面のデータ表示を制御する
- サブ画面からメイン画面の関数実行を制御する
- サブ画面からメイン画面の受注登録を実行し、Channelを使用したプッシュ表示を実現する
- Pull Taskでバックアップ（SQL -> NoSQL）

Dig4 Cloud First + UX3連携 + PullTask

① デスクトップPCの登録処理

キー	タイプ	製品名	価格	CPU	メモリ	OS
itemMas(PC0001)	デスクトップ PC	e-space・・	66800	Core i5 650	4GB	Windows 7 Home

② 液晶ディスプレイの登録処理

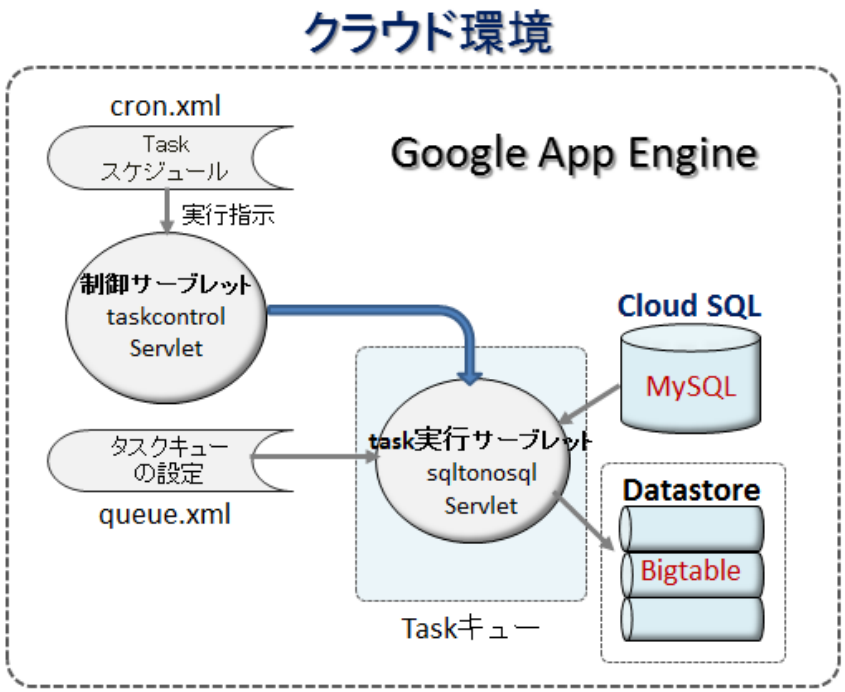
キー	タイプ	製品名	価格	サイズ
itemMas(PC0002)	液晶ディスプレイ	Flat-Wide・	16800	21 インチ

③ プリンタの登録処理

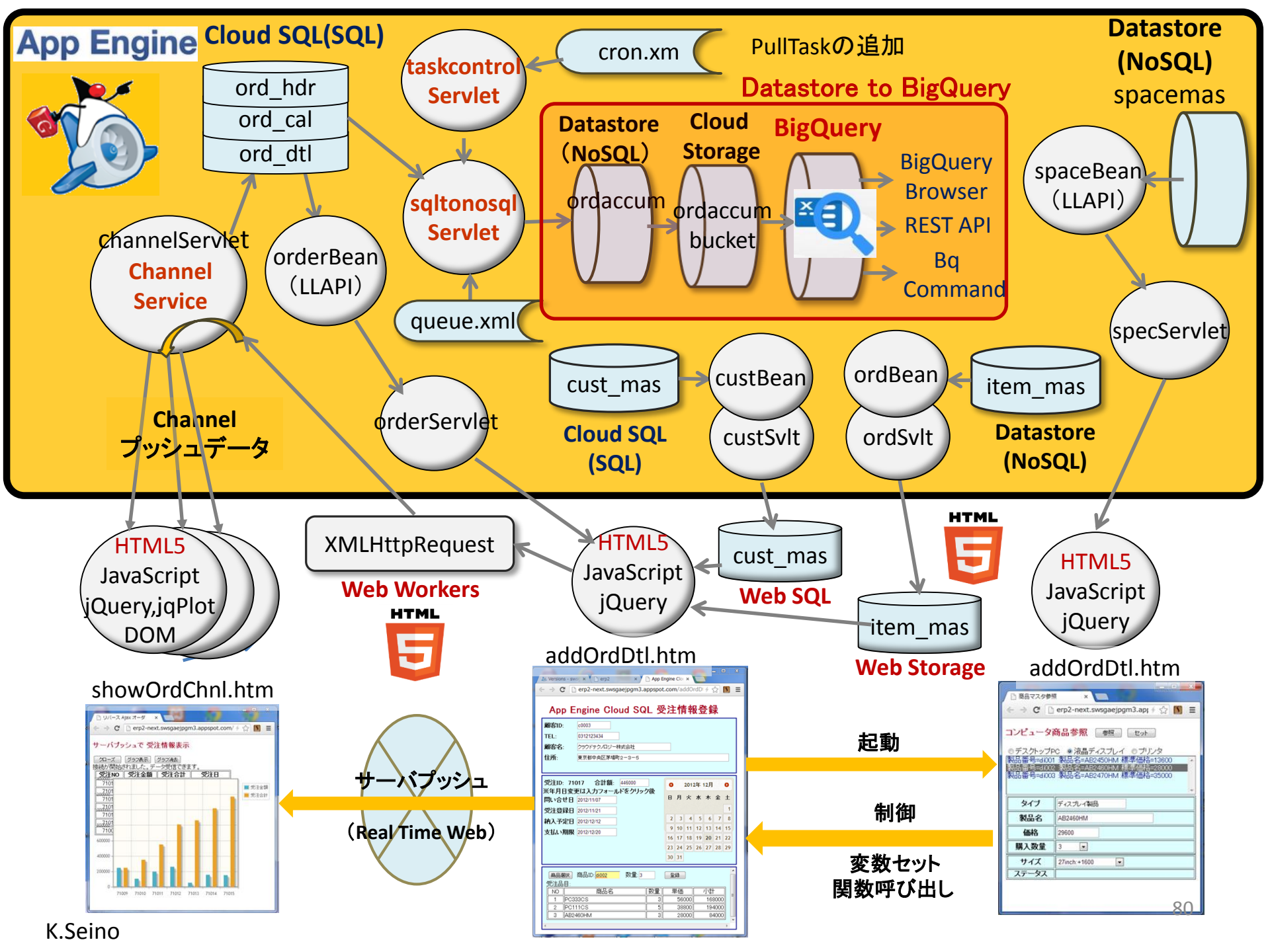
キー	タイプ	製品名	価格	用紙サイズ	解像度
itemMas(PC0003)	プリンタ	Inkjet。。	24800	A4	9600x2400bps

インデックス	タイプ	製品名	価格	CPU	メモリ	OS
itemMas(PC0001)	デスクトップPC	e-space・・	66800	Core i5	1GB	Windows 7 Home
				Core i7	2GB	Windows7 Pro
				Core2 Quad	4GB	
					8GB	
					16GB	

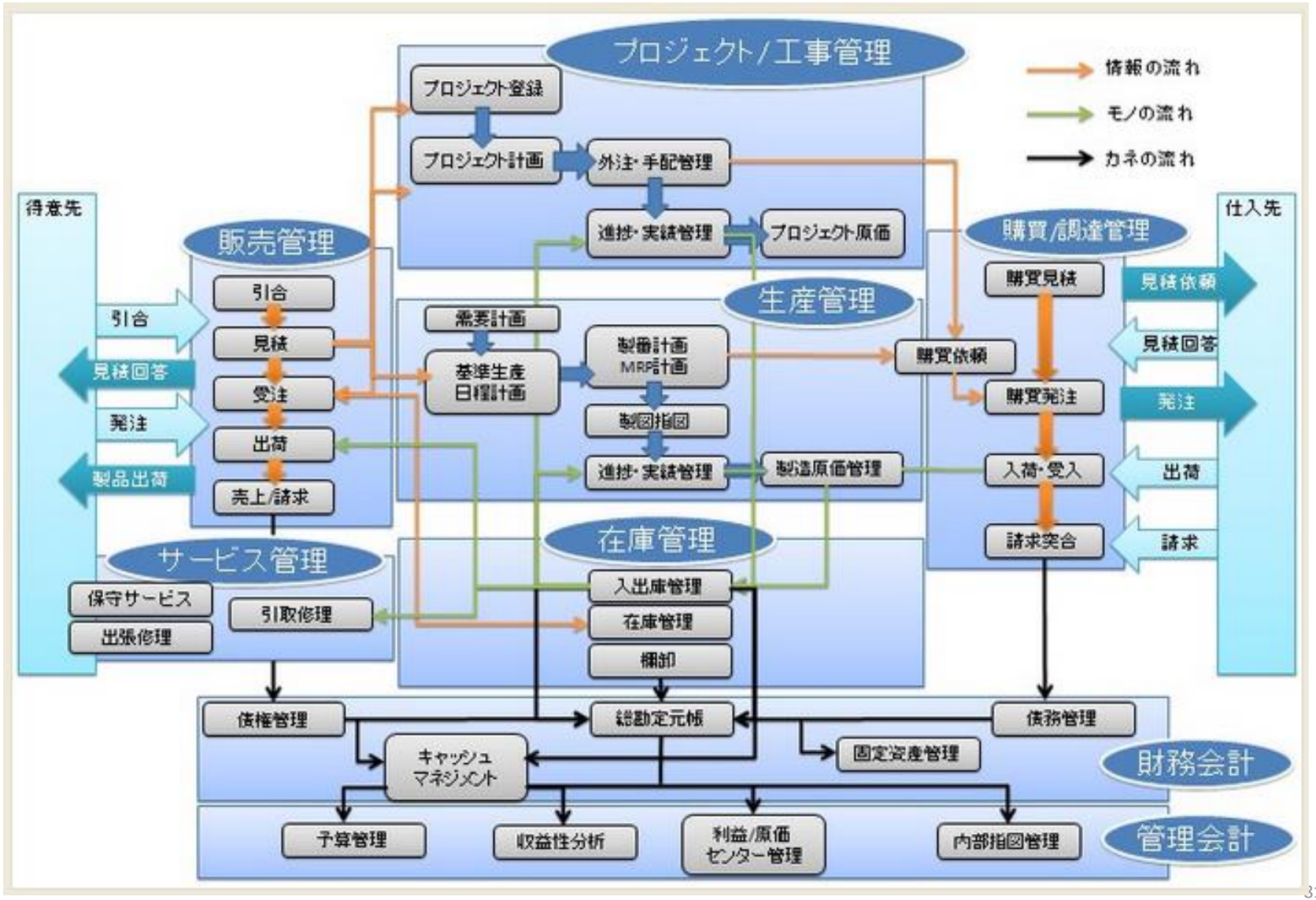
任意に拡張



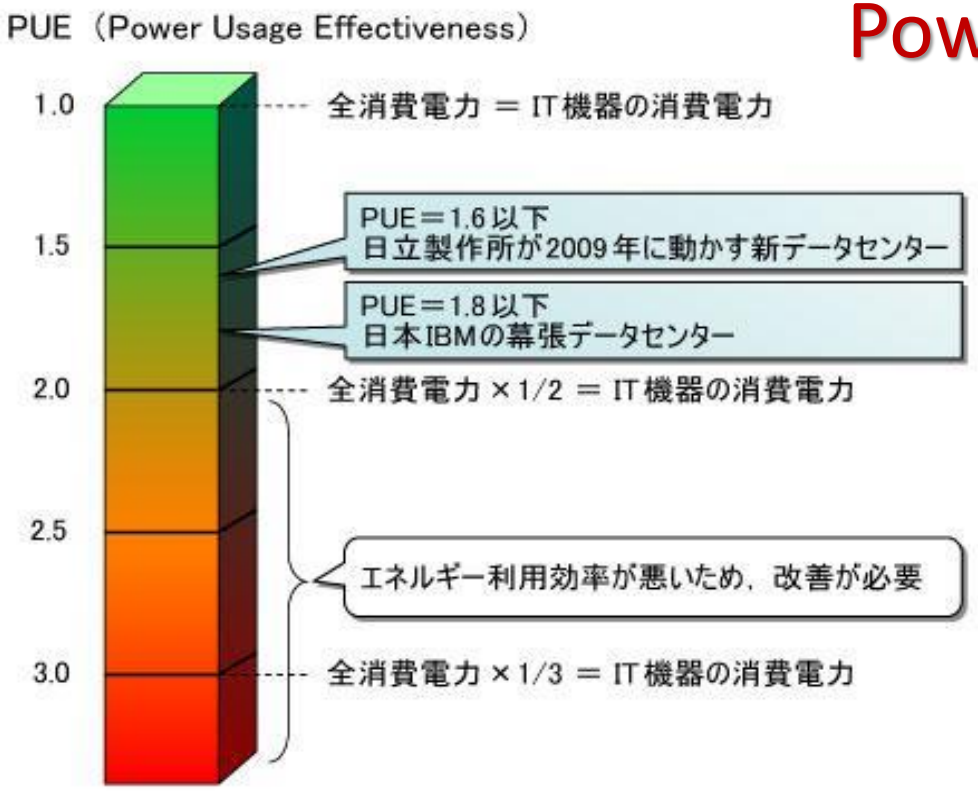
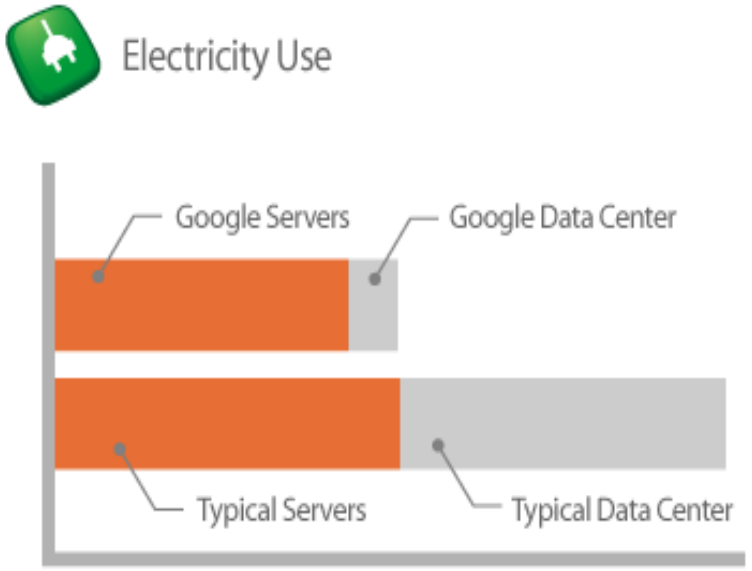
79


App Engine
Cloud SQL(SQL)
ord_hdr
ord_cal
ord_dtl
taskcontrol Servlet
cron.xm
PullTaskの追加
Datastore to BigQuery
Datastore (NoSQL)
spacemas
Datastore (NoSQL)
Cloud Storage
BigQuery
ordaccum
ordaccum bucket
BigQuery Browser
REST API
Bq Command
sqltonosql Servlet
spaceBean (LLAPI)
channelServlet
Channel Service
orderBean (LLAPI)
queue.xml
specServlet
cust_mas
custBean
custSvlt
ordBean
ordSvlt
item_mas
Cloud SQL (SQL)
Datastore (NoSQL)
Channel プッシュデータ
orderServlet
HTML5 JavaScript jQuery,jqPlot DOM
XMLHttpRequest
Web Workers
HTML5 JavaScript jQuery
cust_mas
Web SQL
item_mas
Web Storage
HTML5 JavaScript jQuery
addOrdDtl.htm
addOrdDtl.htm
showOrdChnl.htm
サーバプッシュ
(Real Time Web)
起動
制御
変数セット
関数呼び出し
K.Seino
80

米国　IaaS、PaaS系クラウドベンダには、業務系・基幹系に対するノウハウが乏しい会社も存在する。国産クラウドでは、前のスライドで紹介したようにこの分野のノウハウをアプリサービスとして提供する。



最後に、クラウド運用の自動化を極限まで追求して、
PUE値の低いデータセンタで運用することが重要になる。



Power Usage Effectiveness