

- Node.js on IBM i -

ナンバープレート読取カメラ と IBM i のリアルタイム連携

Open Source 協議会 IBM i

2019年5月24日

KMGホールディングス
九州三菱自動車販売株式会社
電算室 開発課マネージャー
稲葉 剛

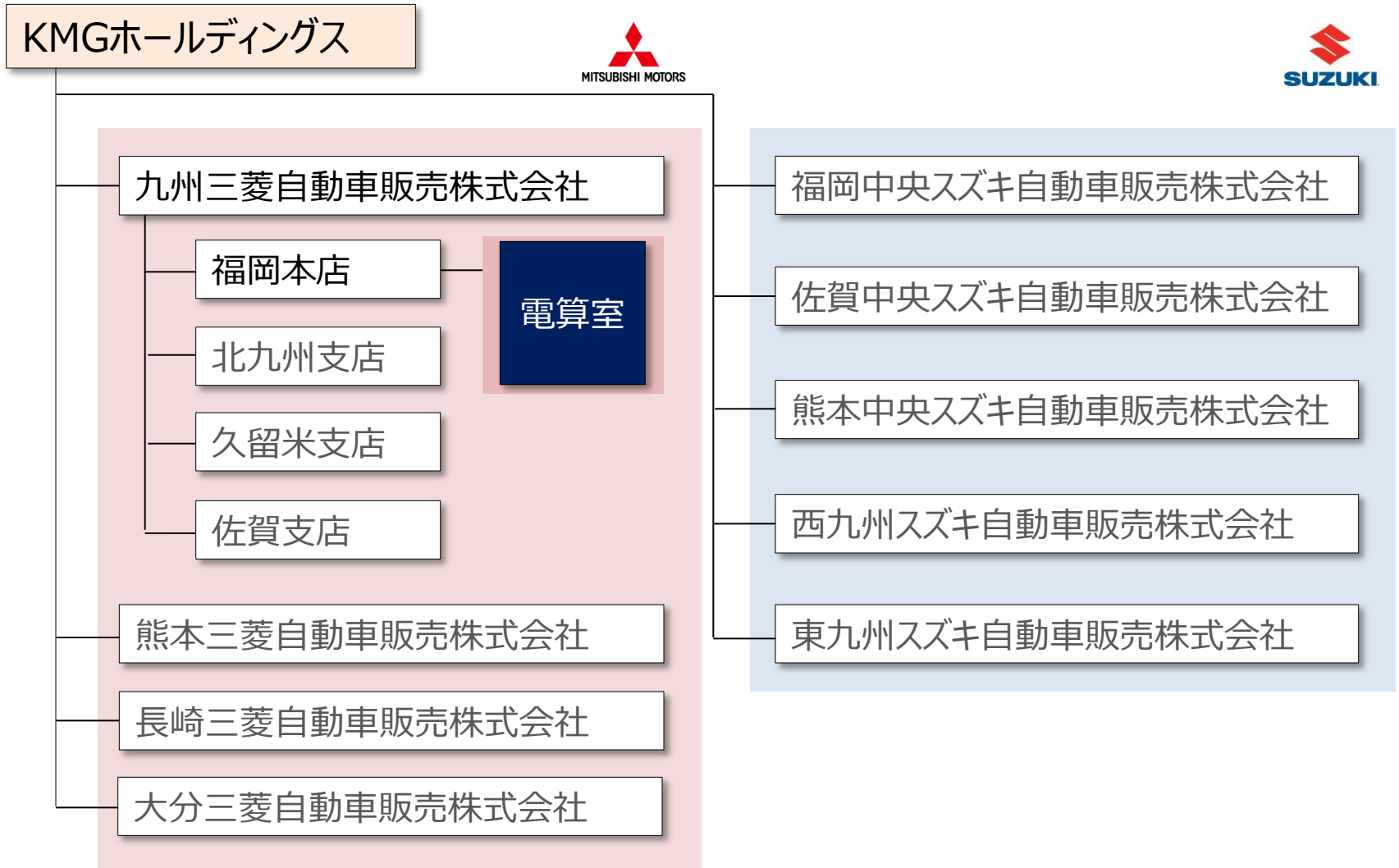
KMGホールディングスの概要

設立年月日	1951年4月2日
資本金	7億5,900万円
従業員数	1,220名
売上高	370億円
グループ会社	9社
販売エリア	福岡・熊本・大分・長崎・佐賀
拠点数	80拠点
事業内容	1. 新車・中古車の販売
	2. 自動車の部品販売
	3. 自動車の修理加工
	4. 保険代理業

(2018年3月期)



KMGホールディングスの組織



目指すソリューション全体像

IBM i と社外システムとのデータ連携

社外システムとデータ連携

IBM i

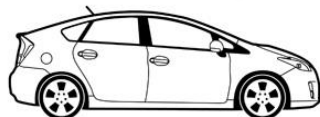


IBM i からクライアントへのプッシュ通知

画像解析ソフトウェア
インストール済 PC

ナンバー認識システム

・福岡 300
た・6-81



IPカメラ

複数PCへ通知可能
画面の最新化は自動

インカムへ
音声通知

画面 + 音声通知



PC音声通知とインカム
音声通知



1. IBM i からクライアントへのプッシュ通知

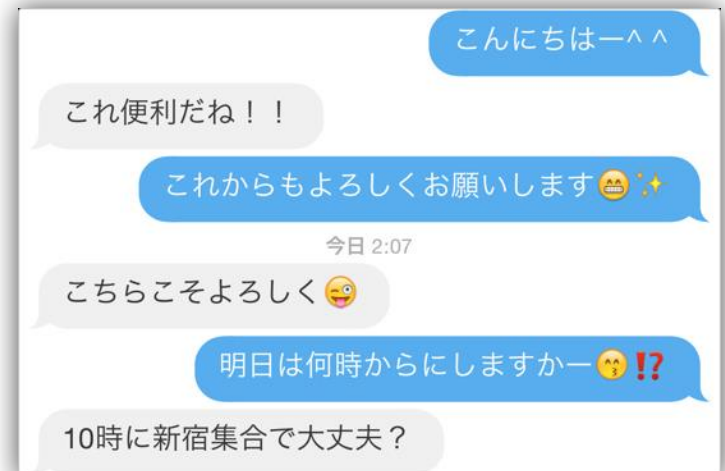
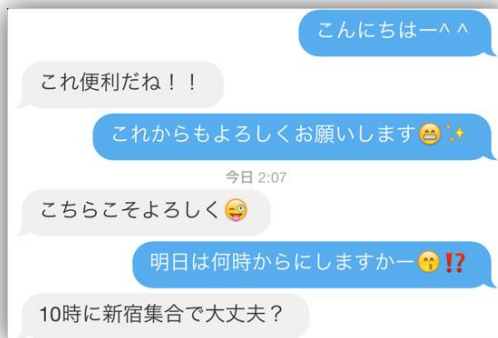
同時に複数画面にプッシュ表示

通知画面は、P C何台でも利用したい
ナンバープレート読み取りデータを受信すると
すべての画面に同じ情報をプッシュ



着想のヒント

チャット画面は何故 自動更新されるのか？



PCでチャットアプリを作って調べよう

サンプルサイトを検索……

必要なのは Node.js

Node.jsのインストール

<https://nodejs.org/ja/>



Node.js® は、Chrome の V8 JavaScript エンジン で動作する JavaScript 環境です。

ダウンロード Windows (x64)

10.15.3 LTS

推奨版

12.3.0 最新版

最新の機能

[他のバージョン](#) | [変更履歴](#) | [API ドキュメント](#) [他のバージョン](#) | [変更履歴](#) | [API ドキュメント](#)

または、[LTSのリリーススケジュール](#)をご覧ください。

公式 Node.js ニュースレター [Node.js Everywhere](#) を購読しましょう。

PCでチャットアプリを作成

<http://www.atmarkit.co.jp/ait/articles/1603/14/news015.html>

WebSocket-Chat

部屋: 名前:

====チャットログ====

====今日の運勢====

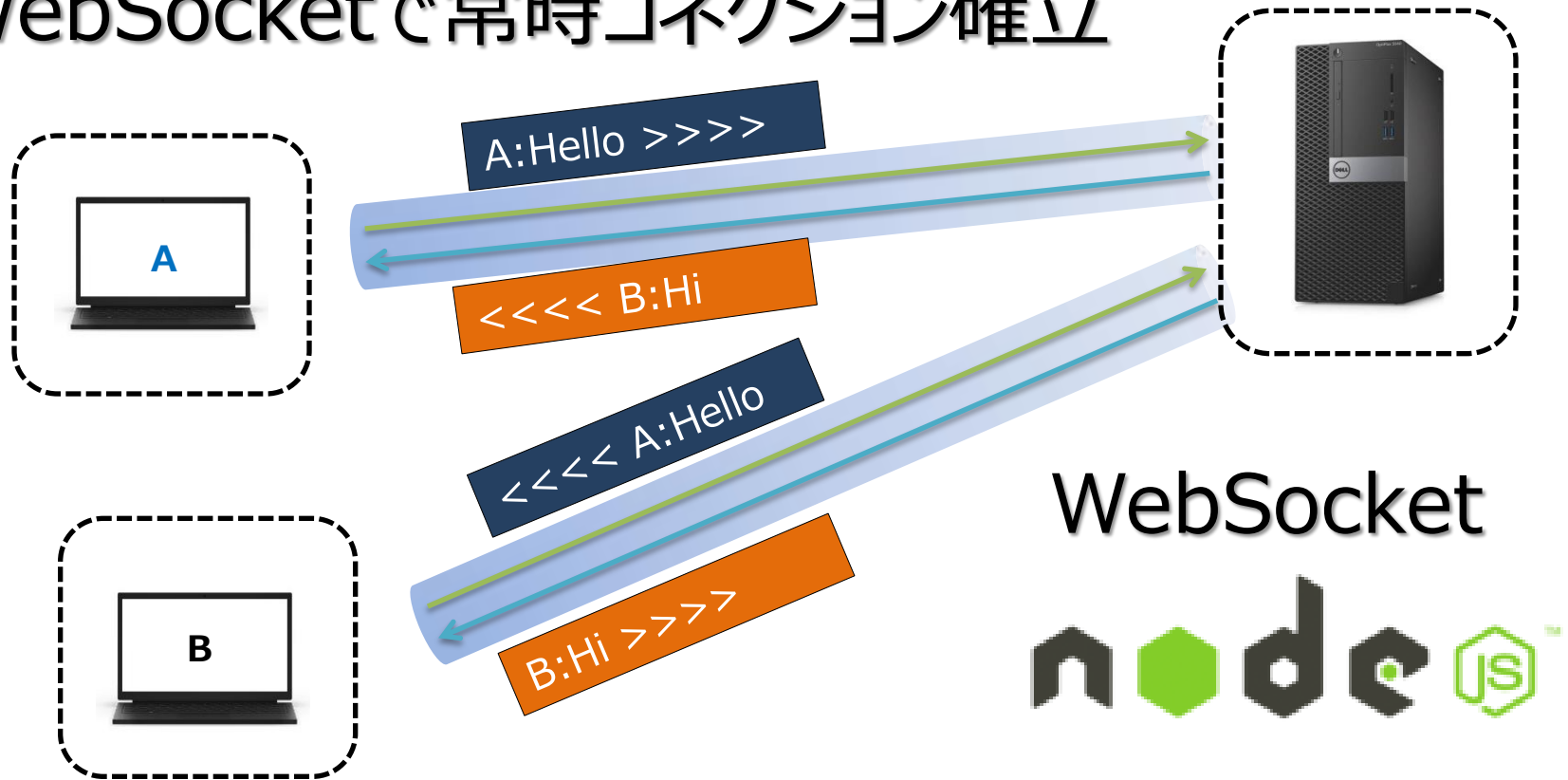
今日のあなたの運勢は... 中吉 です。

確認できたこと
ルームログイン機能
プッシュ表示
複数画面間チャット

チャットってどういう仕組み？

リアルタイムWeb

WebSocketで常時コネクション確立



Node.js とは？

単独でWEBサーバーとして動作

＞ Apacheなどで 別途 Webサーバーを構築する必要なし

サーバーサイドで動作するJavascript

＞ クライアントサイドの Javascriptとほぼ同じ文法なので
学習コストが低い



Node.js とは？

機能拡張は、npm (オープンソースライブラリー) からインストール

> 複雑な機能をもったアプリケーションを短期間で作成可

ノンブロッキング I / O

> 非同期処理により、出来る処理から行うことで高速に動作
小さな処理を大量に行う目的に非常に適しています



IBM iでも Node.jsが使える

Node.js が IBM i でサポートされます

1 NODE.JS とは？

2014年10月に発表された IBM i 7.1 TR9 と IBM i 7.2 TR1 の機能拡張の中に、『5733-OPS』という見慣れぬプログラム番号がありました。

製品名称は『Open Source for IBM i』とあり、オープンソースソフトウェアを IBM i に拡張するためのライセンスプログラムであり、2014年10月に Node.js、2015年5月の IBM i 7.1 TR10 と IBM i 7.2 TR2 の発表の際にプログラミング言語の Python のサポートが追加されています。

さて、Node.js とは何かと言うと、サーバーサイド JavaScript の 1 つです。



IBM i Node.js



IBM iへのNode.jsインストール

前提条件

IBM i V7R1以降

必須ライセンス

【5770-SS1】

option33 – Portable Application Solutions Environment (PASE)

【5733-SC1】

option1 – OpenSSH, OpenSSL, zlib functions, IBM Portable Utilities for i

【5770-DG1】

*BASE – IBM HTTP Server for I

【5733-OPS】

OPT1 IBM I OpenSource Solutions



| IBM i (AS/400)のPASE環境でnode.js



IBM iへのNode.jsインストール

Group PTF

SF99368 Level31 以降

参考サイト

IBM i (AS/400)のPASE環境でnode.jsを動かす方法 [準備編] (1/4)

© 2015/8/21 技術系



Hello World!!

Hello.js サンプルコード

```
var http = require('http');
var server = http.createServer (
function (request, response) {
    response.writeHead(200, {'Content-Type': 'text/plain; charset=utf-8'});
    response.write('Hello World!!');
    response.end();
}).listen(10082);
```



Node.js 起動

CCSID:5035のセッションで以下を実行

>call qp2term

```
選択項目またはコマンド  
===> call qp2term
```

カレントを変更後にnodeプログラムを実行

>cd /QOpenSys/QIBM/ProdData/OPS/Node6/bin

```
===> cd /QOpenSys/QIBM/ProdData/OPS/Node6/bin
```

>node hello.js

```
===> node hello.js
```

ブラウザで確認

<http://<server address>:10082>

DB2アクセスについて

```
1 var http = require('http');
2 var db = require('/QOpenSys/QIBM/ProdData/OP5/Node6/os400/db2i/lib/db2a');
3 var conn = new db.dbconn();
4 var year = new Date().getFullYear();
5 var month = ("0"+(new Date().getMonth() + 1)).slice(-2);
6 var day = ("0"+new Date().getDate()).slice(-2);
7 var ymd = year + month + day; //今日の日付
8 //console.log(ymd);
9
10 http.createServer(function (req, res) {
11     var kyo = '49';
12     var sql1 = "SELECT RSU106,RSU1Z4 FROM PODBLIB.DESU1 WHERE RSU102="
13         + "'" + kyo + "'" + "AND RSU104 = " + "'" + ymd + "'" + " ORDER BY RSU1Z4 DESC";
14     conn.conn("*LOCAL");
15     var stmt = new db.dbstmt(conn);
16     stmt.prepareSync(sql1);
17     stmt.executeSync();
18     var arr0 = new Array;
19     stmt.fetchAllSync(function callback(rs1){
20         arr0=rs1;
21     });
22
```

DB2 Driver 読込

DB2 読込
ブロッキング・ノンブロッキング
の書き方ができる
← ブロッキングの書き方

Toolkitサービスも利用可能 (Node.jsから RPG/CL CALL)

```
var http = require('http');
var db = require('/QOpenSys/QIBM/ProdData/OP5/Node6/os400/db2i/lib/db2a');
var xt = require('/QOpenSys/QIBM/ProdData/OP5/Node6/os400/xstoolkit/lib/itoolkit');
var conn = new db.dbconn();
var conn_r = new xt.iConn("*LOCAL");
```

itoolkit Driver 読込

```
var request;
var response;
```

```
http.createServer(function (req, res) {
  request = req;
  response = res;
  //console.log(arr0);
  var pgm = new xt.iPgm("CESU50", {"lib":"SOPGMLIB"});
  pgm.addParam("4C");
  pgm.addParam("130014259");
  pgm.addParam("20171108");
  pgm.addParam("215000");
  conn_r.add(pgm.toXML());
  conn_r.run(function (rsp) {
    console.log(rsp);
    var results = xt.xmlToJson(rsp);
    results.forEach(function(result, index){
      result.data.forEach(function(data, index2){
        console.log("type:" + data.type + " value:" + data.value);
        delete conn_r;
      });
    });
  });
});
```

RPG CALL

2. IBM i と社外システムとのデータ連携

WebSocket Chat System



Server

常時接続

常時接続

Client

Client

画面 1

画面 2

自分

福岡123あ1234

画面 1

福岡123あ1234

送信

送信

Chat interface for Client 1 (画面 1). It features a blue speech bubble containing the text "福岡123あ1234" with the label "自分" (self) above it. At the bottom, there is a white input field and a blue "送信" (Send) button.

Chat interface for Client 2 (画面 2). It features a blue speech bubble containing the text "福岡123あ1234" with the label "画面 1" (screen 1) above it. At the bottom, there is a white input field and a blue "送信" (Send) button.

ステップ1
Serverで情報変換

登録番号から
顧客マスターとマッチング



Server

常時接続

常時接続

画面1

Client

画面2

Client

自分

三菱 太郎様
エクリプスクロス
車検日 7/3

画面1

三菱 太郎様
エクリプスクロス
車検日 7/3

送信

送信

ステップ2
Clientからのインプットを
カットインで行う

登録番号から
顧客マスターとマッチング



Server



福岡123あ1234

Client

画面1

自分

三菱 太郎様
エクリプスクロス
車検日 7/3

送信

常時接続

Client

画面2

画面1

三菱 太郎様
エクリプスクロス
車検日 7/3

送信

基幹システムとのデータ連携

ナンバープレート読取システムには、読み取ったデータをテキスト化して、任意のフォルダに書き出す機能がある

＞ IBM iのIFSフォルダをネットワークドライブとして登録する

画像解析ソフトウェア
インストール済PC

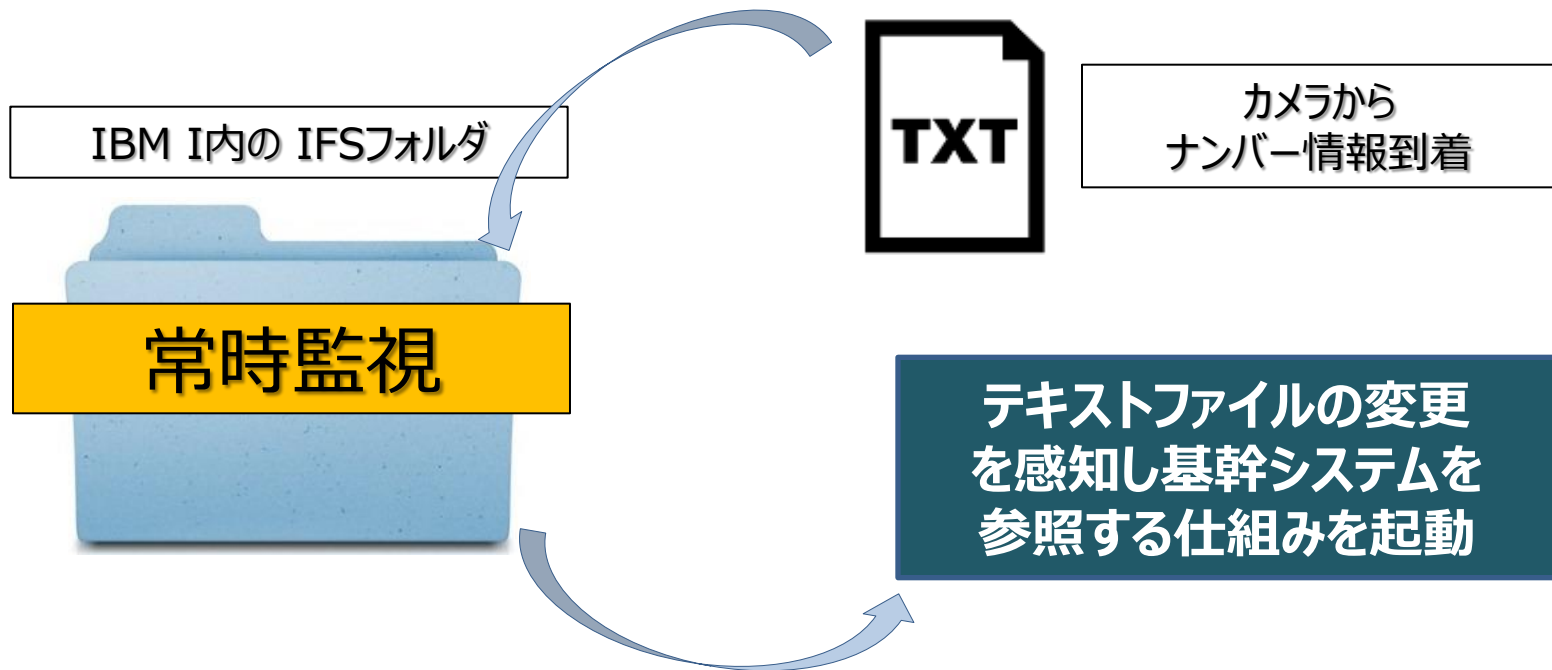
ネットワークドライブに登録
ex (G:)ドライブ

ライブラリ
ドキュメント
IBM i IFS
統合ファイルシステム

_svn
0_Competition
1_Operation
2_WinTIPS
_2007
_2008

フォルダ監視の仕組みが必要

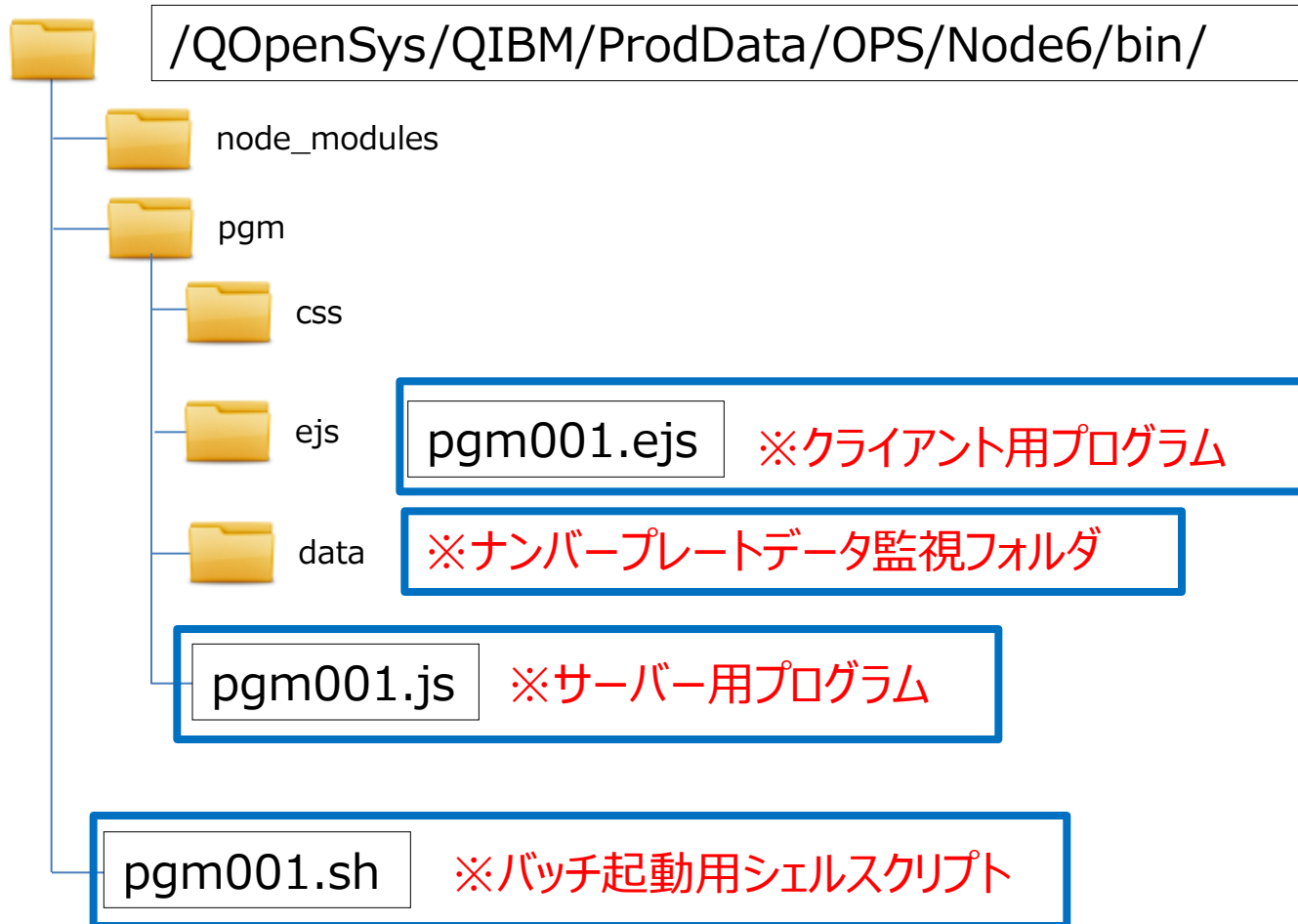
基幹システムのデータ参照をキックする為には
テキストファイル到着を常時監視する仕組みが必要



Node.jsの拡張機能 npm

Node.jsの拡張機能を使えば
フォルダ監視が可能

プログラム配置



バッチ起動用シェルスクリプト

```
#!/usr/bin/sh
```

```
-- バッチからNodeを起動する際のpathの設定等を実行 --
```

```
export QIBM_MULTI_THREADED=Y
```

```
export PATH=/QOpenSys/QIBM/ProdData/OPS/Node6/bin:$PATH
```

```
export LIBPATH=/QOpenSys/QIBM/ProdData/OPS/Node6/bin:$LIBPATH
```

```
-- サーバー用プログラムの起動 --
```

```
node /QOpenSys/QIBM/ProdData/OPS/Node6/bin/pgm/pgm001.js
```

サーバー用プログラム

Node.jsで記述 プログラム処理フロー

① 利用モジュール群の読込

```
const http = require('http'); // ネットワーク基本モジュール
const socketio =
require('/QOpenSys/QIBM/ProdData/OPS/Node6/lib/node_modules/socket.io');
const db = require('/QOpenSys/QIBM/ProdData/OPS/Node6/os400/db2i/lib/db2a');
const xt = require('/QOpenSys/QIBM/ProdData/OPS/Node6/os400/xstoolkit/lib/itoolkit');
```

② WEBサーバー起動 > 初期画面表示

```
var server = http.createServer(getFromClient);
server.listen(8124); //ポート8124
```

③ 起動したサーバーにソケットを関連付ける (WebSocketの有効化)

```
var io = socketio.listen(server); //socket待受
```

サーバー用プログラム

Node.jsで記述 プログラム処理フロー

④ 3つのソケットを準備

◆ LOGIN状態監視

どのルームにだれが入室中か

◆ ハートビート

接続が切れていたら再接続を試みる

◆ フォルダの監視

テキストデータが到着したら、次のプログラムを起動

テキスト内容を読み込み

ToolkitでIBM i から関連データを取得（顧客名・修理履歴・予約情報等）

して、返却データをクライアントへおくる（Socket : server_to_client）

クライアント用プログラム

テンプレートエンジン EJS (Embedded Javascript Templates) を導入
call qp2term > npm install -g ejs

※EJSの導入により、画面プログラムをHTMLの作法で記述できるようになる

```
<!DOCTYPE HTML>
<!-- 登録番号読取システム -->
<html>
<head>
<title><%=title %></title>
<link rel="stylesheet" href="./css/format.css" type="text/css" />
<link rel="stylesheet" href="./css/nds00010.css" type="text/css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<!-- C01. Socket.IOクライアントライブラリの読み込み -->
<script type="text/javascript" src="/socket.io/socket.io.js"></script>
</head>
<body style="font-family:'メイリオ',Meiryo;">
```


クライアント用プログラム

HTML作法で記述 プログラム処理フロー

① 初期画面を表示

通知画面に 現在までに読取した車輛の明細を表示
サーバー用プログラムが、IBM i のDBを読み、対象データをEJSにレンダリングする

② 新しいテキストデータが到着した場合

サーバー用プログラムが、テキストデータを読込、IBM i でToolkitによる RPG CALL
Tooikitの結果セットが、Socket : server_to_client でクライアントへ受け渡される

クライアントで受け取ったデータは直ちに、Jqueryで 1 明細分をテーブルに挿入される

クライアント側 通知画面

来店通知		- 本日のご来店台数 19台 通知回数 21回 -			2019年5月18日(土)		
来店	車両情報	お客様情報	用件	予定	お知らせ	担当	受付
15:08	74007 SK82TM72 デリカトラック 初年度:2009/03	福岡三菱自動車販売株式会社 九州三菱本店 社用車 福岡市中央区薬院 車検 2020/03/25 法点 2019/03/25			前回入庫 2019/03/10 車検		
<div style="border: 2px solid blue; padding: 5px; display: inline-block;">  新しい通知情報を挿入 (プッシュ表示) </div>							
15:04	[Redacted]	[Redacted]			前回入庫 2019/02/01 一般	[Redacted]	
15:01	[Redacted]	[Redacted]					
14:55	[Redacted]	[Redacted]	一般	15:00	前回入庫 2019/05/13 車検交換	[Redacted]	[Redacted]
14:23	[Redacted]	[Redacted]					
	7591#6242	[Redacted]			前回入庫 2019/04/20 一般	[Redacted]	[Redacted]

3. PC音声通知とインカム音声通知

音声変換について

画面表示用のデータ と 音声出力用データを作成

Javascript - Speech Synthesis API で音声変換

```
var speaker = null;  
speaker = "オプソースキョウキカイ化-エムアイ";  
var u = new SpeechSynthesisUtterance();  
u.text = "";  
u.text = speaker;  
u.lang = 'ja-JP';  
u.rate = 1.1;  
speechSynthesis.speak(u);
```

EJSに記述
PCから発話

発話用情報のデータ変換

- 漢字のままだと、読み間違いが多いので半角カナに変換
- 数字は連続すると“センニヒャクサンジュウヨン”と読まれるので [] で括る
- 句点をいれることで、一呼吸おいた発話になる

発話用変換データ

業務連絡です。
登録番号 [1] [2] [3] [4]
カハシ様ご来店です。
シャケンのご予約があります。
担当営業はサカトです。
お出迎えをお願いします。

聞こえるイメージ

業務連絡です
登録番号 1 2 3 4
高橋様ご来店です
車検のご予約があります
担当営業は坂本です
お出迎えをお願いします

インカム連携

PCの音声出力とインカム音声ユニットを連携することで
割込み通知を実現

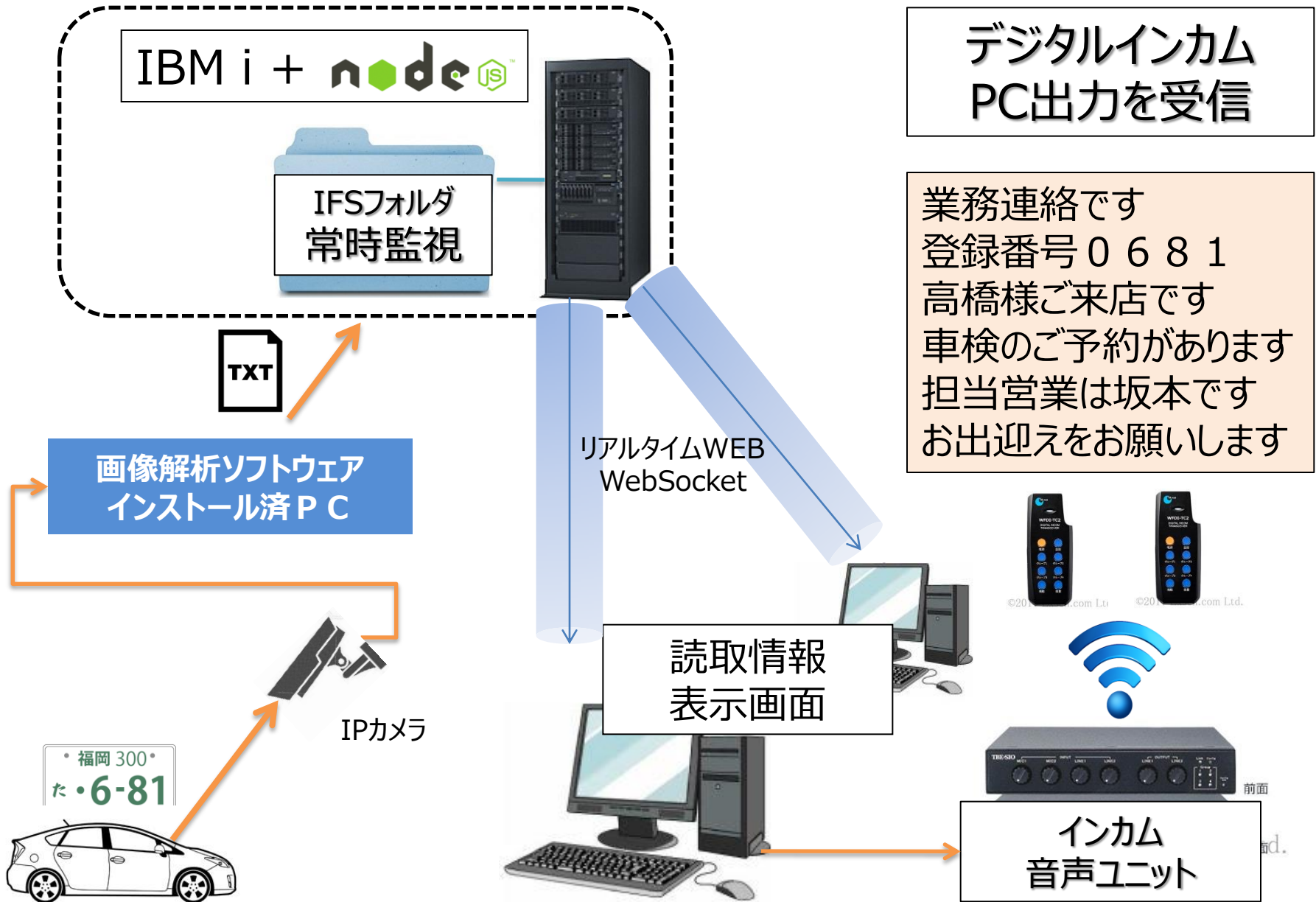


デジタルインカム
PC出力を受信



完成したシステム全体図

システム全体図



デジタルインカム
PC出力を受信

業務連絡です
登録番号0681
高橋様ご来店です
車検のご予約があります
担当営業は坂本です
お出迎えをお願いします



インカム
音声ユニット

Open Source 協議会 IBM i

ありがとう
ご褒美に