

DARWIN Architectureと Google Prediction API

グループI4

2008MI233 鈴木健太

2008MI214 沢田天馬

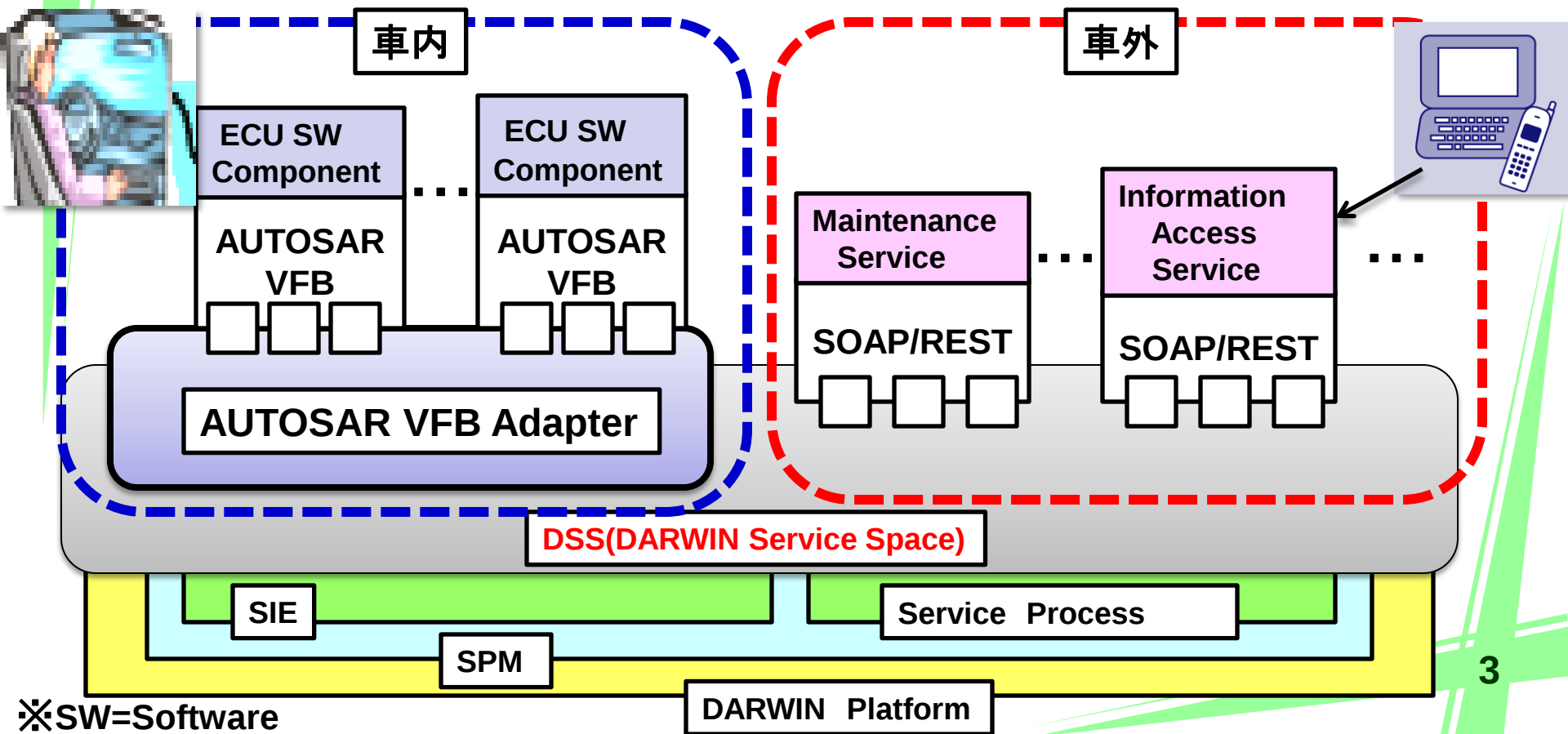
目次

- **DARWIN Architecture**
- **DARWIN Service Spaceについて**
- **DARWIN Service link Protocol**
- **Google Prediction API**
- **Prediction APIの実装**
- **今後の課題**
- **参考文献**

DARWIN Architecture

サービス/コンポーネントの統合/相互運用性を提供する

- ECUソフトウェアコンポーネント
- DSS上にあるAUTOSAR VFB Adapter
- サービスリンクプロトコル(後で詳細を参照)



DSS(DARWIN Service Space)

・ DSSに求められる課題

自動車アプリケーション用の高安全性、信頼性、リアルタイムのパフォーマンスの要件を満たしている分散オブジェクト技術でないといけない
ちなみにJavaspacesやCORBAなどの既存の技術では満たされていない

重要な機能

1,クラウドベースのDSSサーバ

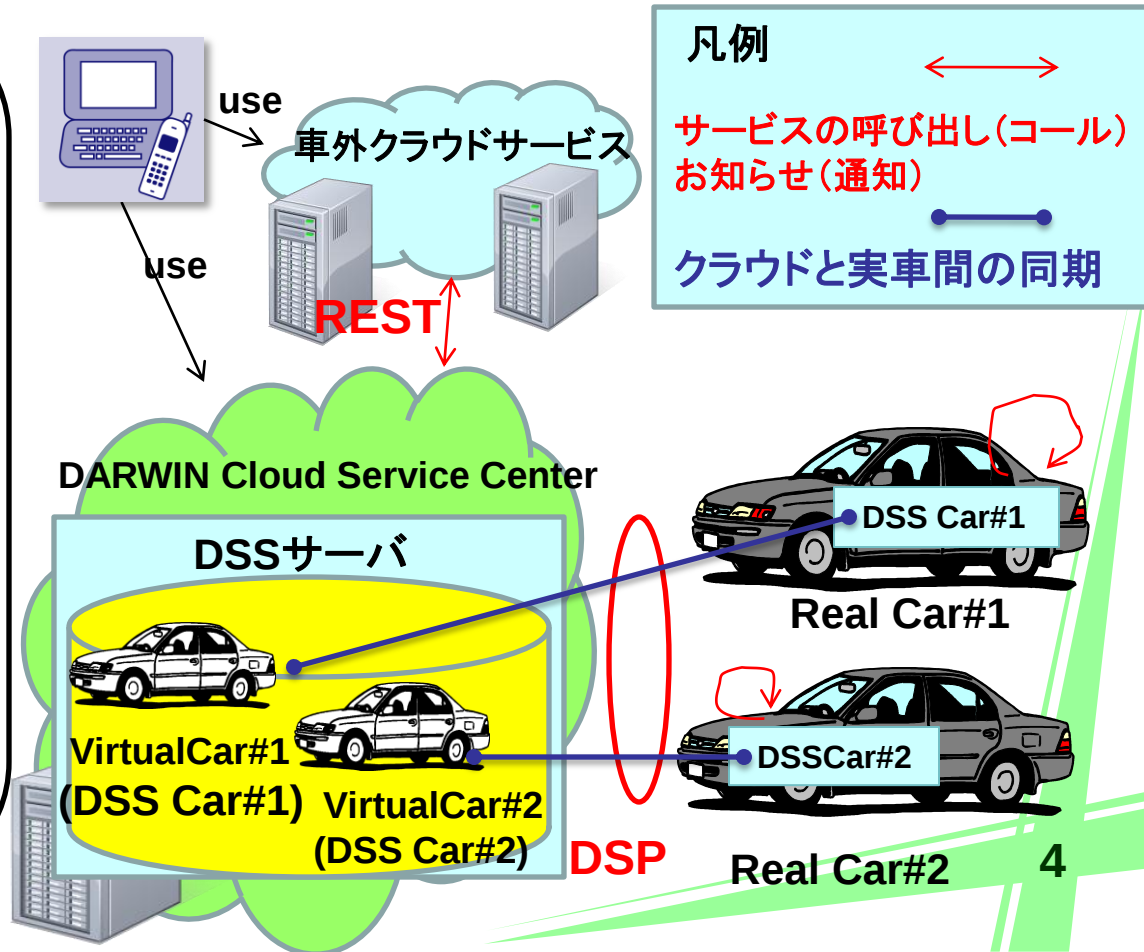
- ・実際の車と車外サービス相互作用のカプセル化

2,AUTOSARVFBのサポート

- ・ECUソフトウェアコンポーネントと他のソフトウェアコンポーネントや車内外サービスとの連携

3, QoS管理メカニズム

- ・事前の優先順位の割り当て



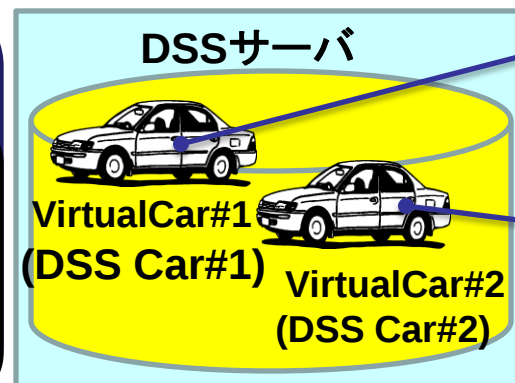
DARWIN Service link Protocol

2種類のサービスリンクプロトコルがサポートされている

1, データセンター間および
車両内の通信について

DSP (Darwin Service Link Protocol)

NGTP (Next Generation Telematics Protocol) のサブセットであり、互換性がある
データ圧縮による高パフォーマンスと
軽量プロトコル



2, データセンターと
車外サービス間の通信について

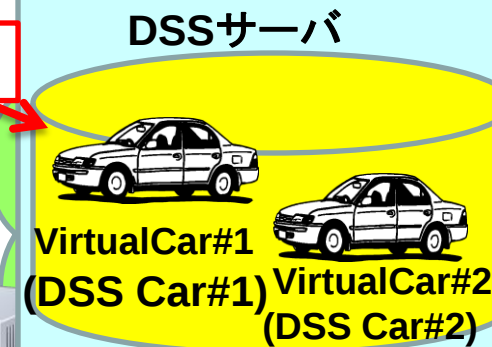
REST (Representational State Transfer)

- ・リソースはURLで識別される
- ・車内・車外のサービスの管理用の拡張機能
CarID, ServiceID, InterfaceIDを用いる

車外クラウドサービス

DARWIN Cloud Service Center

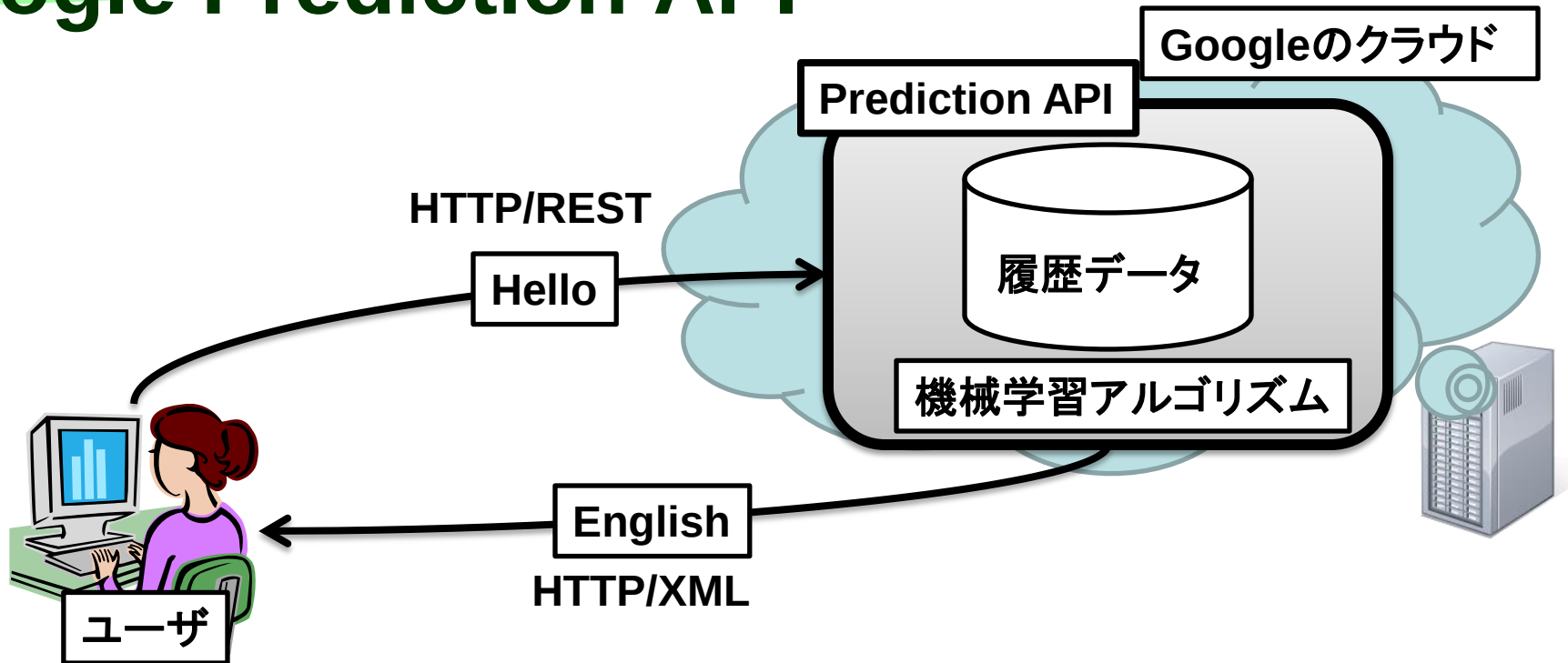
REST



URLのサンプル

<http://darwin.jp/api/001/vehicle/{car ID}/services/{service ID}/interfaces/{interface ID}/call>

Google Prediction API



特徴

RESTfulなAPIであり、RESTを用いてやり取りをおこなう

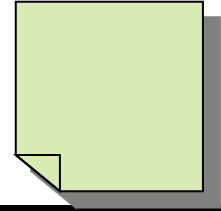
数値データ、非構造化データを扱うことができる

Google App Engineやコマンドラインなど多くのプラットフォームからアクセス可能

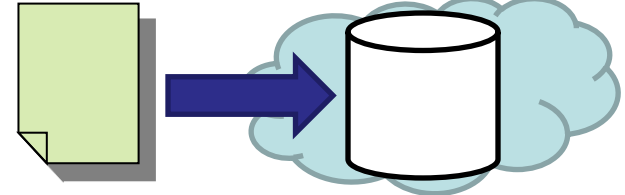
APIの使い方の流れ

7

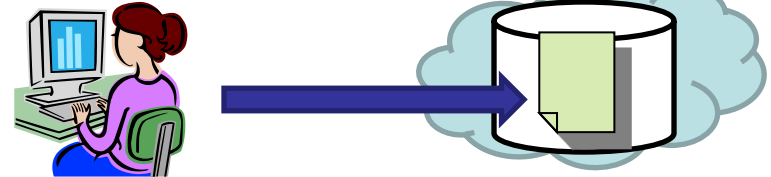
- 1, 機械学習に必要な
トレーニングデータを用意する



- 2, Google Storageに
トレーニングデータをアップロードする

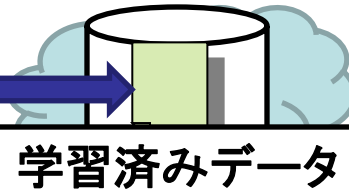


- 3, Prediction APIのメソッドを
トレーニングデータに対して呼び出す

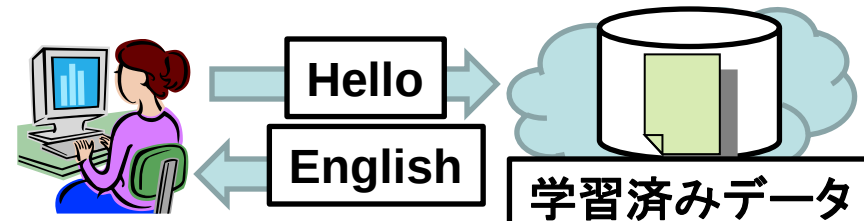


- 4, モデル構築

Googleの
機械学習アルゴリズム



- 5, Predictionクエリを送ると
最適な予測結果が返ってくる



組み込みとの連携の可能性

Fordの取り組み

運転履歴をトレーニングデータとした
Prediction APIの利用

クラウドを利用した**大規模な計算処理**
豊富なストレージ

自動車の走行の効率化

いつ・どこで・どれだけ運転したか



次回以降の運転時

予測される目的地
最適なルートを提供



Prediction APIの実装 1/4

Hello Predictionの実装

- 1, 機械学習に必要な
トレーニングデータを用意する

今回は、Googleが用意してくれている”language_id.txt”を利用

今回のHello Prediction では、スペイン語・フランス語・英語を利用可能



language_id.txtの中身も3か国の言葉が羅列されていた

Prediction APIの実装 2/4

2, Google Storageに
トレーニングデータをアップロードする

Google Storageは、コマンドラインやGUIを用いて、様々なデータを保存可能

bucketと呼ばれる自分の領域を作成
・ユニークな名前でないといけない
今回は(kenta08mi233)とした。



先ほどのlanguage_id.txtをアップロード

Prediction APIの実装 3/4

3, Prediction APIのメソッドを
トレーニングデータに対して呼び出す

4, モデル構築

シェルスクリプトを用いて, storage内のlanguage_id.txtに対してリクエストを送る



trainと呼ばれるもので、予測のためのモデルを構築する

Googleが提供するSample:oauth-train.sh

```
#!/bin/bash
# Train a prediction model.
# Usage: oauth-train.sh MODEL_NAME

ID=$1
KEY=`cat googlekey`

post_data="{\"id\": \"$ID\"}"

# Train the model.
java -cp ./oacurl-1.2.0.jar com.google.oacurl.Fetch -X POST \
-t JSON \
"https://www.googleapis.com/prediction/v1.2/training?key=$KEY" <<< $post_data
echo
```

今回はここでつまってしまった。

Prediction APIの実装 4/4

実装ができない考えられる理由

シェルスクリプトがWindows環境では実行できないため、Linuxで行わなくてはならない

サンプルのoauth-train.shの中身がよくわかっていないため、リクエストエラーがでてしまう



A terminal window screenshot showing a command execution and its output. The terminal title bar indicates the user is '08mi233' on host '08mi233.G30'. The menu bar shows 'ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)'. The command executed is `./oauth-train.sh kenta08mi233/language_id.txt/languages`. The output is a JSON error object: `{ "error": { "errors": [ { "domain": "global", "reason": "badRequest", "message": "Bad Request" } ] }, "code": 400, "message": "Bad Request" }`. The prompt shows the user is still in the `08mi233` shell.

```
08mi233@08mi233.G30
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
08mi233@08mi233 08mi233]$ ./oauth-train.sh kenta08mi233/language_id.txt/languages
{
  "error": {
    "errors": [
      {
        "domain": "global",
        "reason": "badRequest",
        "message": "Bad Request"
      }
    ],
    "code": 400,
    "message": "Bad Request"
  }
}
08mi233@08mi233 08mi233]$
```

今後の課題

- Prediction APIの実装をやり遂げること
- 車とクラウドという狭い範囲でなく、組み込みとクラウドというもう少し広い範囲で課題を探していく

参考文献

- **Service-Oriented Architecture for
Automotive Cloud Service Systems**
Akihito Iwai, Mikio Aoyama
- **自動車ネットワークサービスのサービス連携アーキテクチャの提案と評価**
水谷拓人
- **車載ソフトウェアのサービスプラットフォームのモデルとアーキテクチャ**
青山幹雄 中道上 佐藤二郎 小島祐次
- **@IT** <http://www.atmarkit.co.jp/>
- **Google Prediction API** <http://code.google.com/intl/ja/apis/predict/>