

**NGTP**



**2008MI011 朝倉知也**

**2008MI079 岩井 大**

# 目次

1. テレマティクスサービスとは
2. NGTPについて
  - 2.1概要
  - 2.2ソフトウェアアーキテクチャについて
  - 2.3システムアーキテクチャについて
3. OSGi - 実装・サービス公開 -
4. 課題
5. 参考文献

# 1. テレマティクスサービスとは

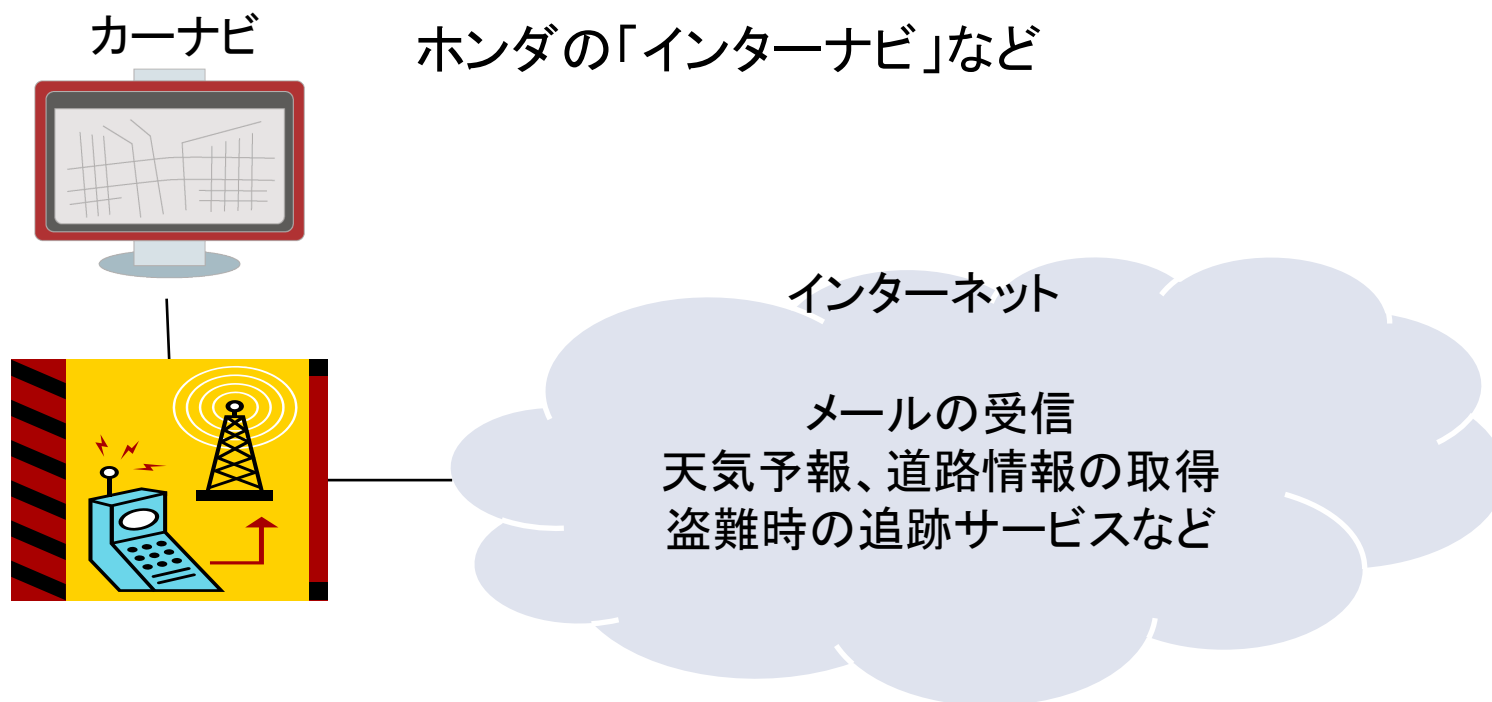
## ○ 自動車向けの次世代情報提供サービス

携帯電話などの端末を利用して自動車をインターネットに接続することで受けられるサービスのこと

(実用例)

トヨタの「G BOOK」、日産の「カーウイングス」

ホンダの「インターナビ」など



## 2.1 NGTP – 概要 –

- 現状のテレマティクスサービスでは、メーカーごとに独自のサービスを提供している  
→単一のサービスプロバイダを使うため**柔軟性**に欠けている



- NGTP(Next Generation Telematics Pattern)  
自動車メーカーが自由に、サービスプロバイダを変更したり、アプリケーションごとに異なるプロバイダを利用したりできる、柔軟性のあるオープンプラットフォームの開発を目的したもの

BMW,Connex,WirelessCarの3社によって開発

2008年1月 Version1.0が公開

現在はVersion 2.0が公開(dispatcher呼ばれる配信機構の追加)

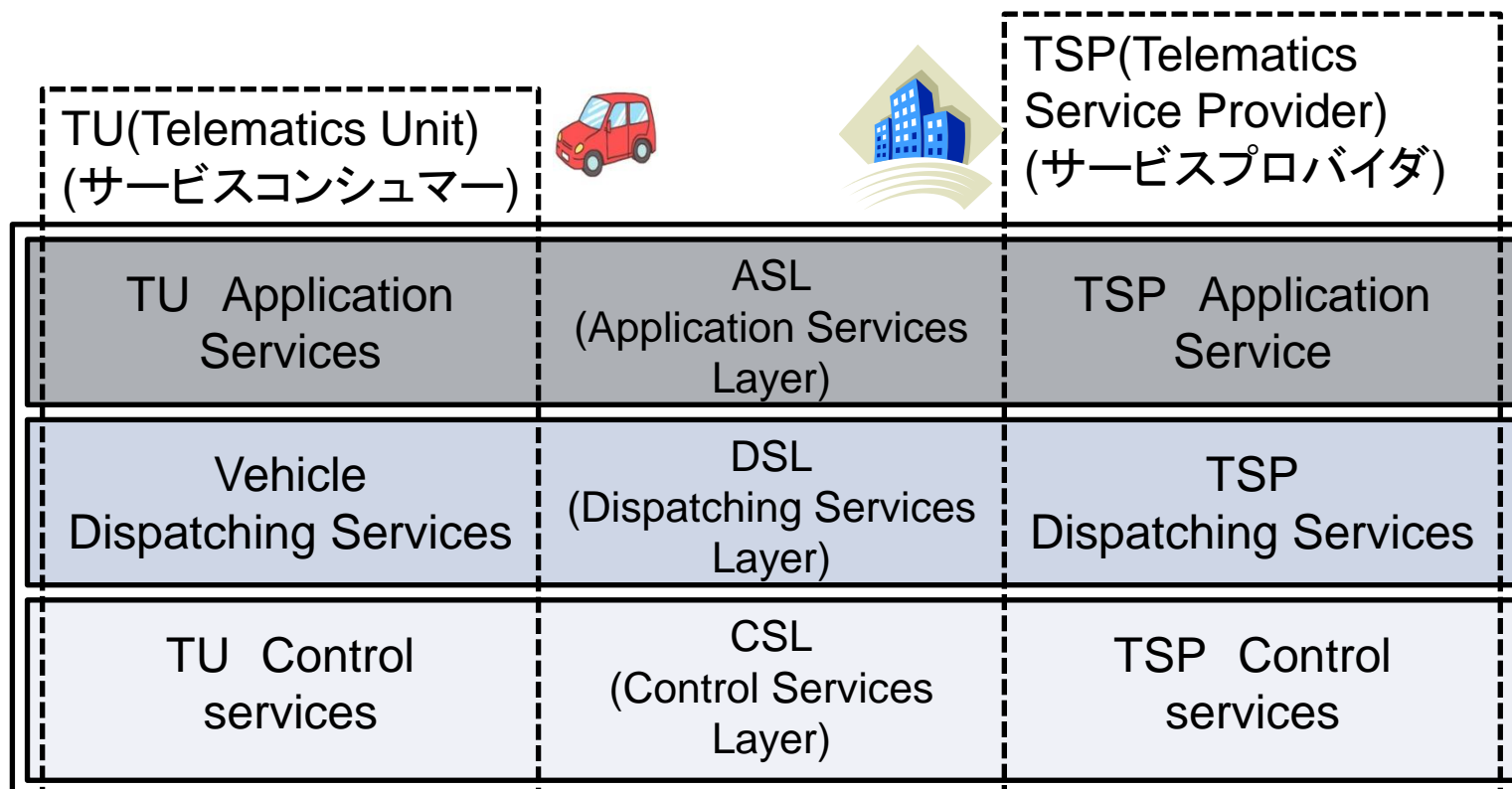
## 2.2 NGTP – ソフトウェアアーキテクチャ –

### ○ ソフトウェアアーキテクチャについて

CSL層・・・セキュリティなどのサービス提供の基盤

DSL層・・・配信先のIDや位置などのサービス配信の Protokol

ASL層・・・個々のサービスを実行するAPI

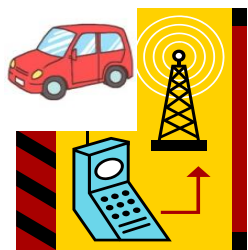
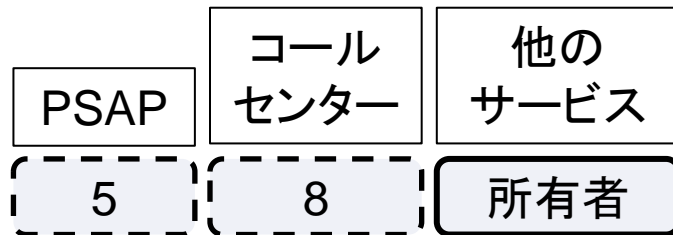


# 2.3 NGTP – システムアーキテクチャ(1/3) –

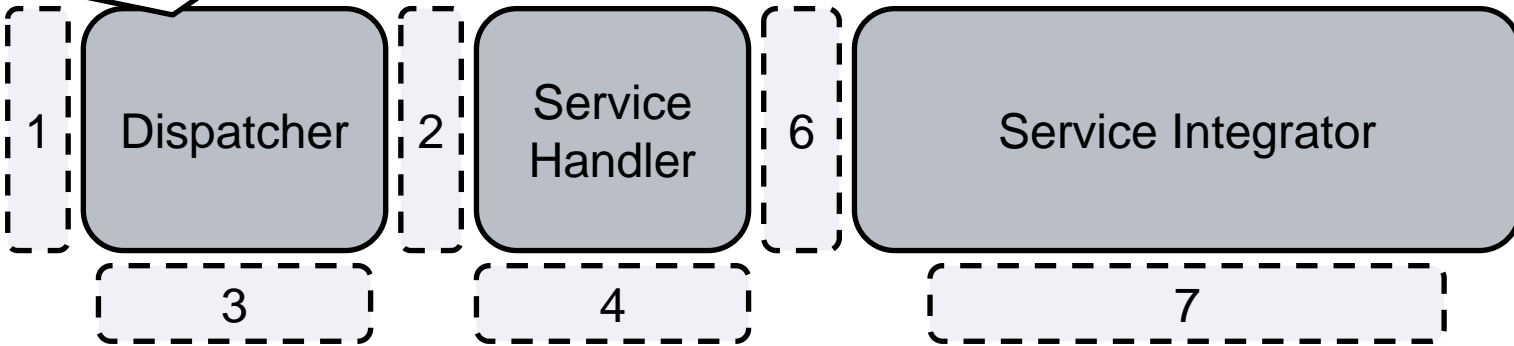
NGTP2.0で用いられるシステムアーキテクチャ

 ……インターフェース

- TUと通信(送受信)
- メッセージを符号・複合化を配信(指示)
- ユースケースの決定



TU

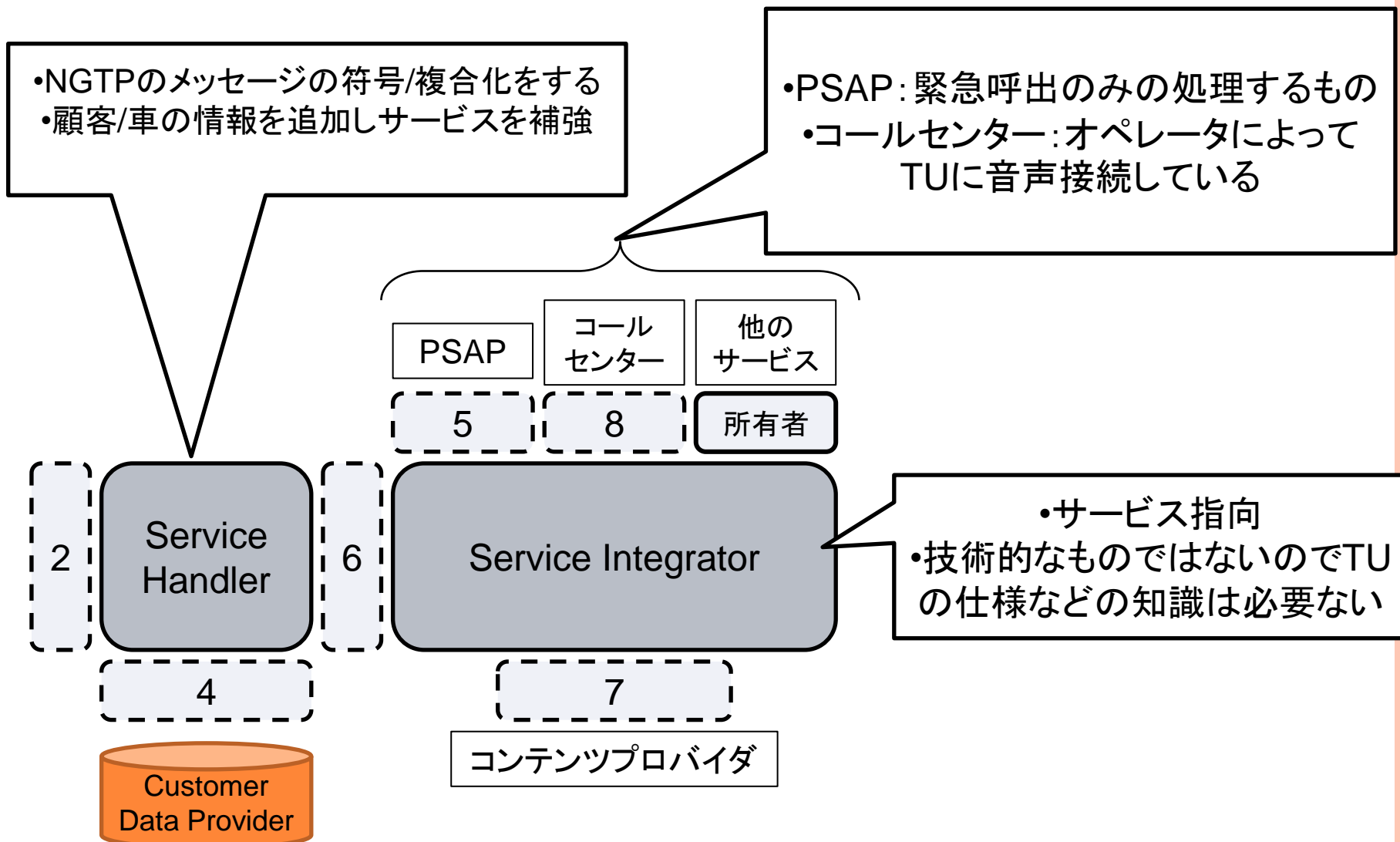


コンテンツプロバイダ

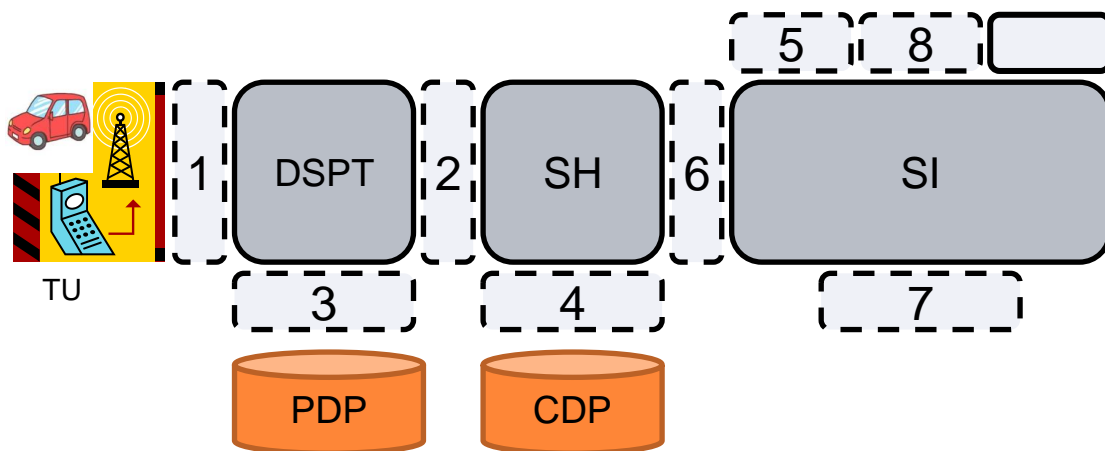
サービスの  
入力パラメータ  
を提供  
(顧客の位置  
サービスの種類)

顧客、車、サービスの情報の管理・配信を行う

## 2.3 NGTP – システムアーキテクチャ(2/3) –



## 2.3 NGTP – システムアーキテクチャ(3/3) –



### インタフェースの役割

- 1.バックエンドとTU間の物理的な通信、双方向通信
- 2.サービスの具体的内容は示されない
- 3.PDPからの情報を特定のメッセージ形式に変換
- 4.CDPからの情報をSHの情報に追加する
- 5.PSAPが処理するe-Callを確立するためのアーキテクチャを可能にするもの
- 6.受信されたサービスのデータにアクセス
- 7.CPにあるコンテンツをDSPT,SHを通じてTUに送信または別のサービスプロバイダに転送
- 8.CCからの情報をSIからコールセンターのソフトウェアに接続



### 3. OSGi – 実装・サービス公開と利用(1/6) –

#### ○ 実装するサービス

与えられた文字列に”\_Response”を付けて返却する.

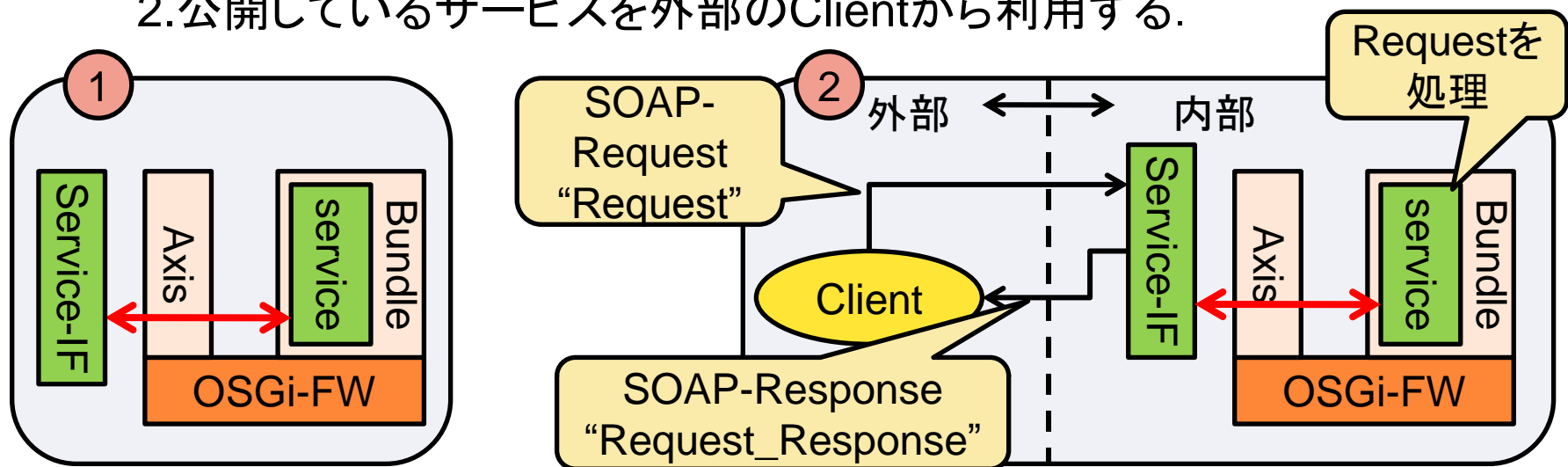
ex) 入力値に”Request” → 返却値:”Request\_Response”

#### ○ 実装手順

1.上記サービスを実装するBundleを作成.

axis-osgi Bundleを利用し, 上記サービスのインタフェースを外部に公開する.

2.公開しているサービスを外部のClientから利用する.



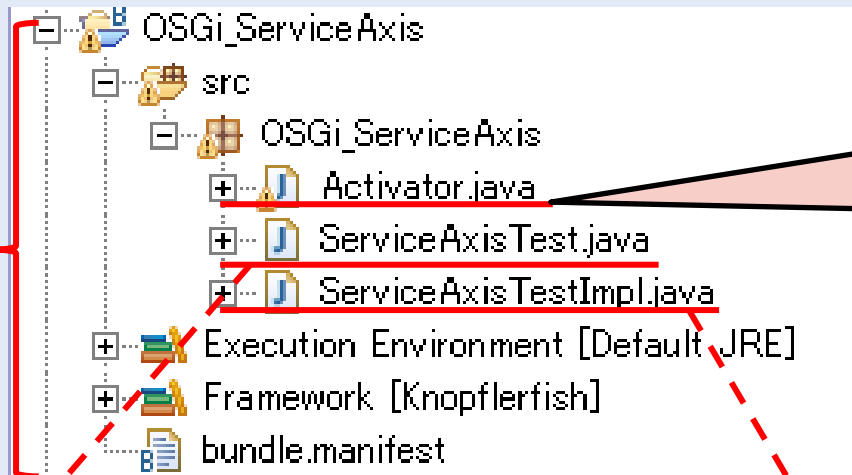
## 3. OSGi – 実装・サービス公開と利用(2/6) –

### 1. サービスを実装するBundleを作成

1-1. 前回, サービス連携で実装したBundleと同様にプロジェクトを作成し, インタフェースの作成とサービス実装を行う.

1-1

作成する  
プロジェクト



Bundleのstart, stopを記述,  
start内でサービスを登録する

インタフェース

```
public interface ServiceAxisTest {  
  
    public String TestRequest(String msg);  
  
}
```

インタフェースの実装

```
public class ServiceAxisTestImpl implements ServiceAxisTest {  
  
    public String TestRequest(String msg) {  
  
        String res = msg + "_Response";  
        return res;  
  
    }  
  
}
```

### 3. OSGi – 実装・サービス公開と利用(3/6) –

1-2.Activator.java – start内で実装したサービスを登録する記述を行う。

1-3.Bundleをエクスポート, Knopflerfishで実行し, 登録したサービスが外部に公開されていることを確認する。

1-2

```
public void start(BundleContext context) throws Exception {

    System.out.println("Start " + getClass().getName());
    test01 = new ServiceAxisTestImpl();
    Hashtable props = new Hashtable();
    props.put("SOAP.service.name", "test01");
    context.registerService(ServiceAxisTest.class.getName(), test01, props);
}
```

1-2-1

サービスのプロパティに  
SOAP.service.nameを指定すると  
Axisを利用し外部にサービスを公開できる

1-2-2

サービスを登録

1-3

http://localhost:8080/axis/services

And now... Some Services

- test01 ([wsdl](#))
  - TestRequest
- axisadmin ([wsdl](#))
  - getPublishedServiceNames

Axisを利用しサービスを公開  
(test01-TestRequestサービスが公開中)  
※axis-osgi BundleがActive状態に  
なっている必要がある

### 3. OSGi - 実装・サービス公開と利用(4/6) -

- 2. 公開したサービスを外部Clientから利用する.
  - 2-1. クライアントを作成する. (Javaで作成, 詳細は省略)
  - 2-2. クライアントを実行し, 挙動を確認する.

2-2

↓ クライアント(test\_SOAPClient3.jar)の実行

```
C:\¥plugins>java -jar test_SOAPClient3.jar localhost 8080 Request
SOAPClient : CONNECT TO http://localhost:8080/axis/services/test01
SOAPClient : SENDMESSAGE IS Request
- 要求されたクラス (javax.activation.DataHandler と javax.mail.internet.MimeMulti
part)が見つかりません。Attachmentサポートが利用できません。 / [en]-(Unable to fi
nd required classes (javax.activation.DataHandler and javax.mail.internet.MimeMu
ltipart). Attachment support is disabled.)
Request_Response
```

送信メッセージを  
引数で指定

接続先  
endpoint

返却値

警告:  
関係ない  
ので無視

## 3. OSGi – 実装・サービス公開と利用(5/6) –

2-3.SOAPメッセージの確認を行う。  
(EclipseのTCP/IPモニターを利用)

2-3

### SOAP-Request

ヘッダ

POST /axis/services/demo HTTP/1.0  
(省略)  
Content-Length: 465

ボディ

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope ~(省略)~>
  (省略)
  <SENDMESSAGE xsi:type="xsd:string">
    Request
  </SENDMESSAGE>
  (省略)
</soapenv:Envelope>
```

送信値

### SOAP-Response

ヘッダ

HTTP/1.0 200 OK  
(省略)

ボディ

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope ~(省略)~>
  (省略)
  <ns1:TestRequestReturn
    xsi:type="xsd:string">
    Request_Response
  </ns1:TestRequestReturn>
  (省略)
</soapenv:Envelope>
```

返却値

## 3. OSGi – 実装・サービス公開と利用(6/6) –

### ○ 実装目的と感想

目的: OSGiでの外部サービスとの連携を行う際に, OSGi-Axisにおける連携方法を理解するため.

感想: 最初に挙げたサービスの実装を行い, 仕様通りに動作することを確認した. また, SOAPメッセージの内容を確認することもできた. しかし, 細かい動作やクライアント側の動作がまだ理解できていない部分もあるため, 理解を進めたい.

## 4. 課題

- 今回はNGTPの概要について調べたが、問題点や具体的な流れを調べていきたい。
- 今後の方針としては、自動車と外部とのサービス連携を軸として、その周辺の課題や解決の方法を模索していきたい

## 5. 参考文献

- IT用語辞典 e-Words(<http://e-words.jp/>)
- NGTP in a nutshell(<http://www.ngtp.org/>)
- NGTP 2.0 Use case examples Version1.0
- 濱千代正弥、片桐雅仁：自動車ネットワークサービスの連携アーキテクチャ
- 水谷拓人：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
- 青山幹雄 ほか：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アーキテクチャ 自動車技術会 学術講演会前刷集
- システムクリエイト株式会社 OSGi入門  
(<http://www.s-cre.co.jp/web/osgi/osgi.php?url=1>)