日本 Apache Camel ユーザ会

「Apache Camel」
適用事例紹介 & 超入門

日本 Apache Camel ユーザ会

古関 伸行

2012年 12月 8日 （土）

発表者

- 古関 伸行（こせき のぶゆき）
- やっていること
  - 以前：
    EAI/BPM製品の展開、導入支援
    SOA関連情報の展開
  - 最近：
    Camelを使って既存商用EAI製品案件勢力図をぬりかえるべく活動中
    （といいつつ、私も Camel 初心者です！！）

発表内容

この発表では、

- Apache Camelはどういったものか？
- Apache Camelをどのように使うか？

について説明します。

目次

- Apache Camelとは？
- Apache Camelの適用事例紹介
- Camel概要
- Camelの使い方
- さらに深くCamelを知る
- Camelコードサンプル

- Apache Camelとは？
- Apache Camelの適用事例紹介
- Camel概要
- Camelの使い方
- さらに深くCamelを知る
- Camelコードサンプル

Apache Camelとは？

Camelを一言でいえば、

- 多数の**コンポーネント**が使える
- **ルーティングエンジン**

です。

…って、意味分かります？

Apache Camelとは？

つまり…

別々のシステム同士や、

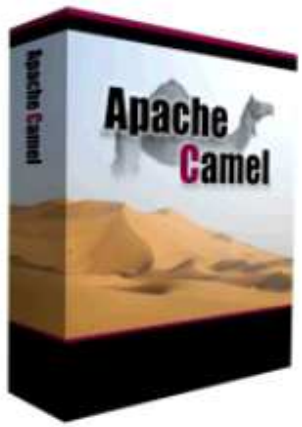
異なるアプリ同士を**つなげて**、

データのやり取りを**簡単に実現**したい

そんなときに役に立つ(かもしれない)ツールです。

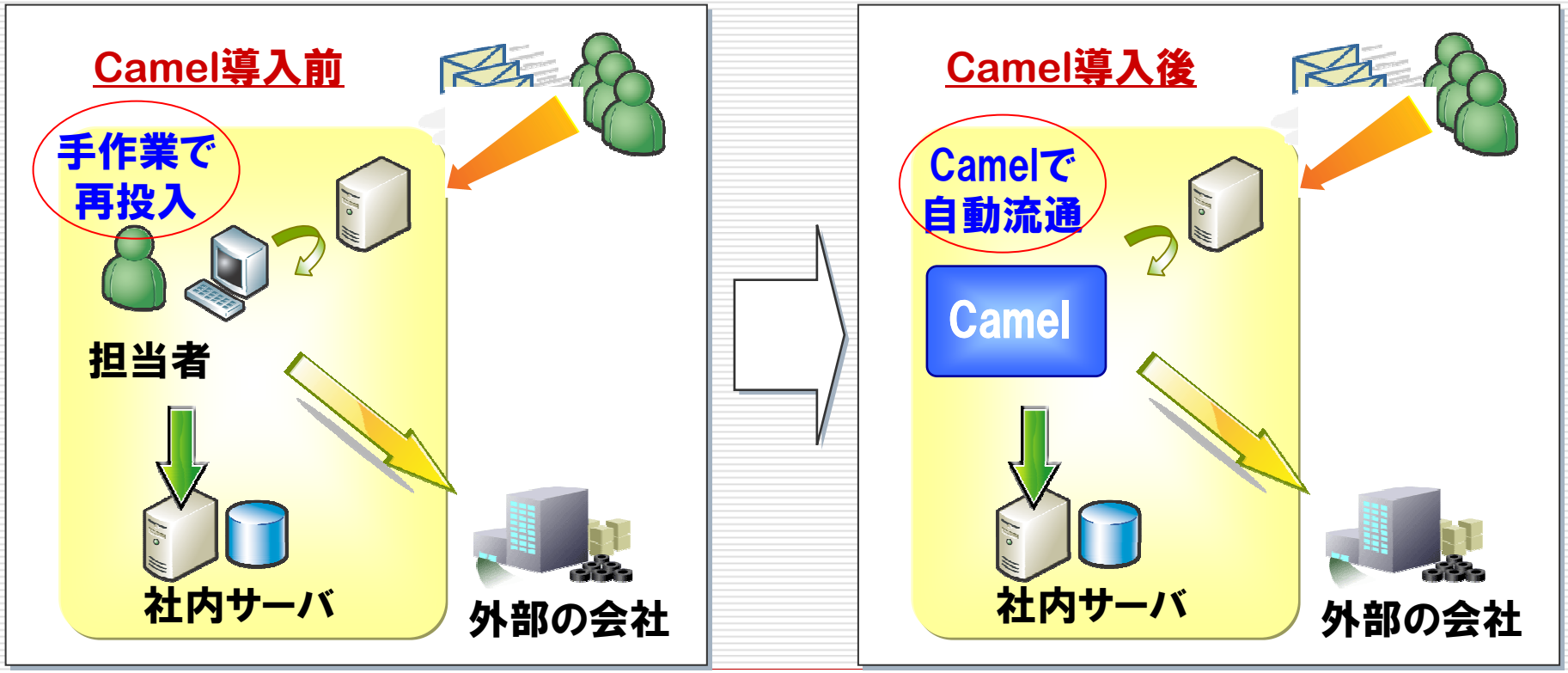
Apache Camelとは？ ～背景～

- Apache Software Foundationのプロダクトの一つ
- 2007年にリリース（ActiveMQのサブプロジェクト）
- 2009年にトッププロジェクトになる
- 数ヶ月毎にバージョンアップ
  →結構活発に活動



Apache Camelとは？ ～何ができるか？ その1～

□ 例えば、手作業を効率化

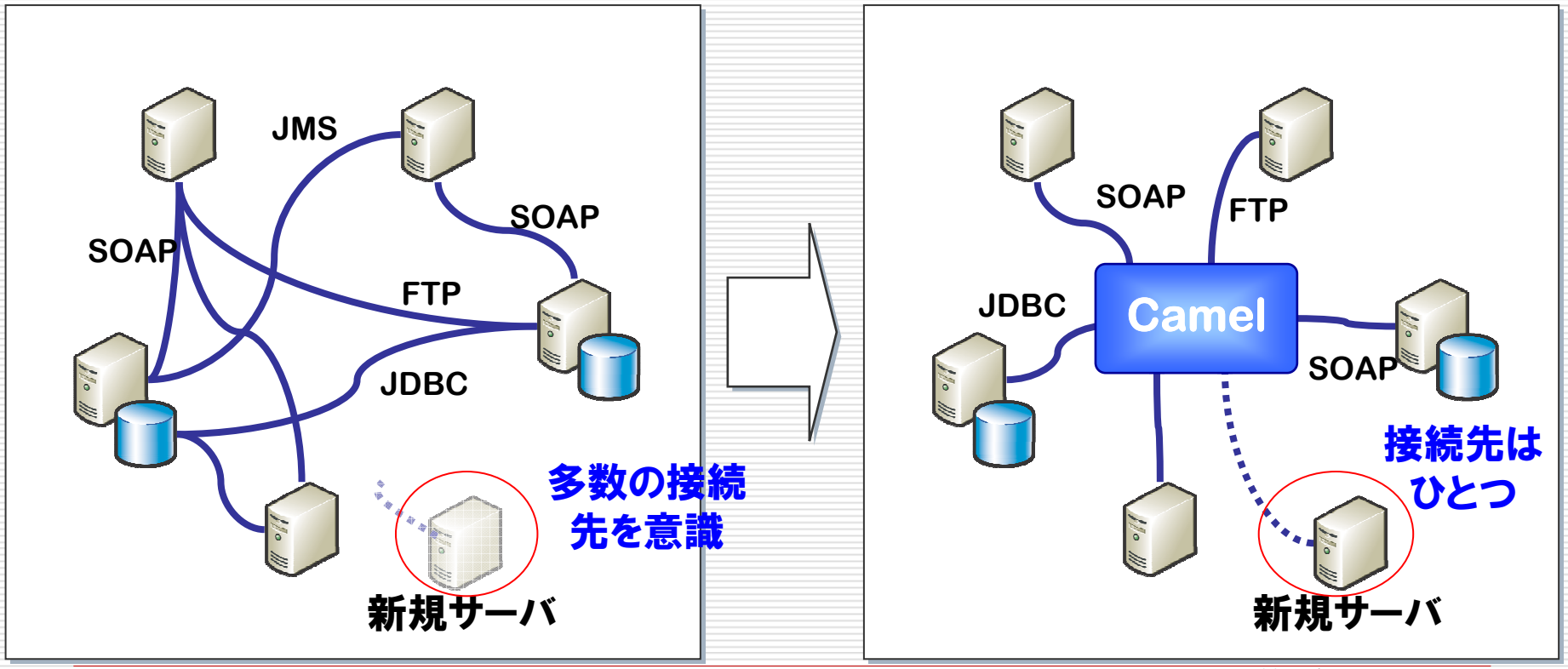


・担当者が手作業で実施

・Camelが自動で実施

Apache Camelとは？ ～何ができるか？ その2～

- 例えば、スパゲティな連携を簡素化



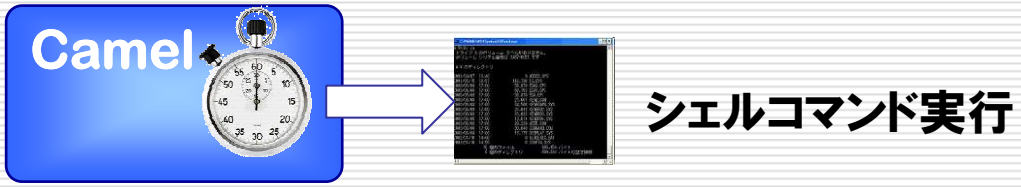
- メンテナンスが煩雑
- 各システムが接続先を意識

- メンテナンスはCamelに集約
- Camelが接続先を振り分ける

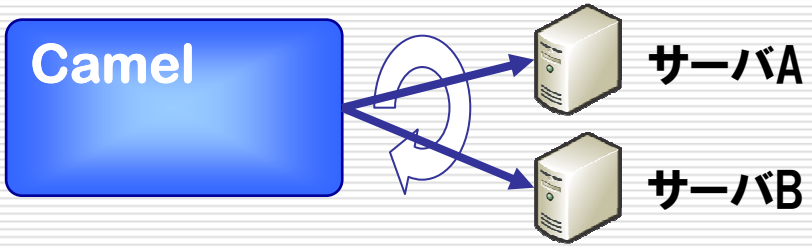
Apache Camelとは？ ～何ができるか？ その3～

- 例えばCamelの部品を利用したら…

一定間隔/決まった日時に特定の処理を実行



ラウンドロビンでの負荷分散



流通データ量の制御

- NW帯域制限
- マシンリソース制限

Apache Camelとは？
～何ができるか？ その4～

- Camelと他の機能を組み合わせると…

◆ Camelをベースに機能拡張

Camel + 機能追加 ⇒ ESBとか

◆ Camelと別機能の組合せ

Camel + BPM ⇒ EAI/BPM

Camel + Rule ⇒ CEPもどき

とっても高価な商用製品と同じようなことを、安価に実現できる可能性

*ESB : Enterprise Service Bus
*EAI/BPM : Enterprise Application Integration / Business Process Management
*CEP : Complex Event Processing

Apache Camelとは？
～まとめてみる～

ロ ルーティングエンジン

- データを右から左にルーティングさせるエンジン
- プロトコルやデータフォーマットの変更も可能

ロ コンポーネント(部品)

- 色んなプロトコルと連携する部品
- データ変換用の部品
- 使い道の不明な部品、等々

- Apache Camelとは？
- Apache Camelの適用事例紹介
- Camel概要
- Camelの使い方
- さらに深くCamelを知る
- Camelコードサンプル

Apache Camelの適用事例紹介

前にも書きましたが…

◆ Camelと別機能の組合せ

Camel ＋ BPM ⇨ EAI/BPM

Camel ＋ Rule ⇨ CEPもどき

この2つの事例について紹介します

Apache Camelの適用事例紹介 ～ その① ～

既にそこそこの規模のシステムで稼動中

- 案件概要
  - システム概要
    オーダの開通設定、開通試験の指示、試験結果情報の更新等を行うシステム
  - システムカバー範囲
    東日本全域
  - 商用EAI/BPMで稼働しているシステムの更改
    ・EAI機能はApache Camelで、
    ・BPM機能はjBoss jBPMでリプレース

Apache Camelの適用事例紹介 ～ その① ～

□ 特色

Apache Camelで複数システム同士を連携

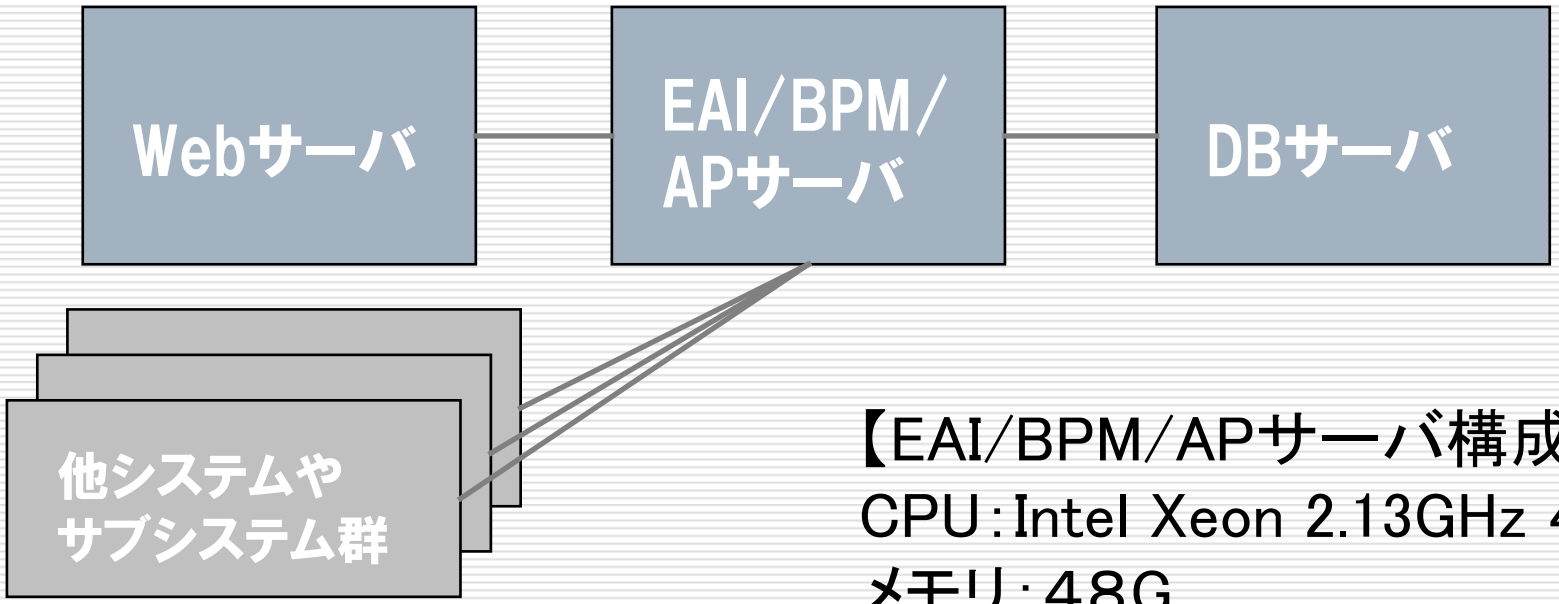
- SOAPで連携
- JDBCで連携（Oracle接続）
- JMS（Oracle Advanced Queue）でjBPMと連携

□ Apache Camelを導入しての感想

- 商用EAI製品と遜色ない生産性、品質
（バグが少なすぎる！！　と、強化試験を実施）

Apache Camelの適用事例紹介 ～ その① ～

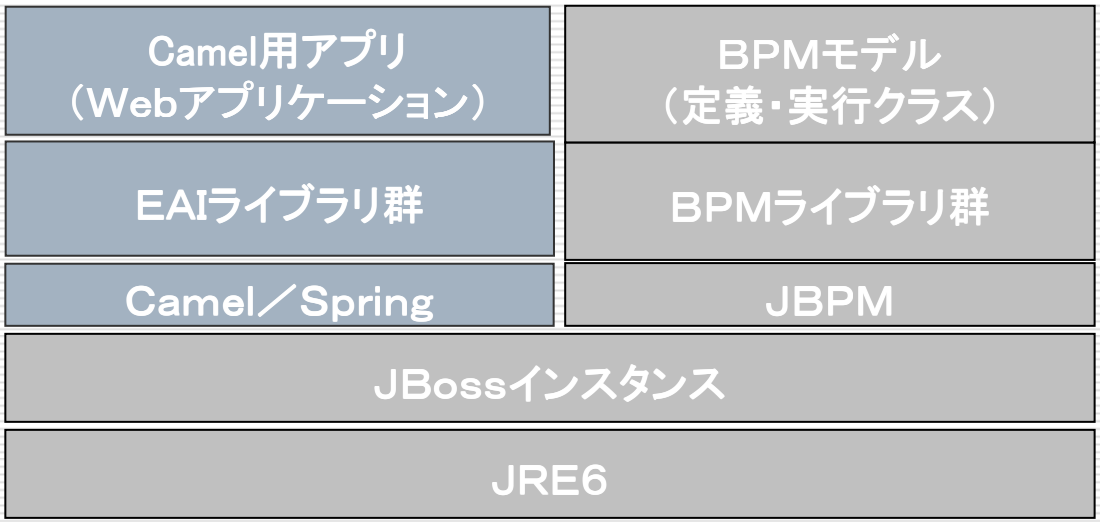
- ハードウェア構成（一部抜粋）
  - APサーバと同居して、同一VM上で動作



【EAI/BPM/APサーバ構成】
CPU：Intel Xeon 2.13GHz 4コア
メモリ：48G

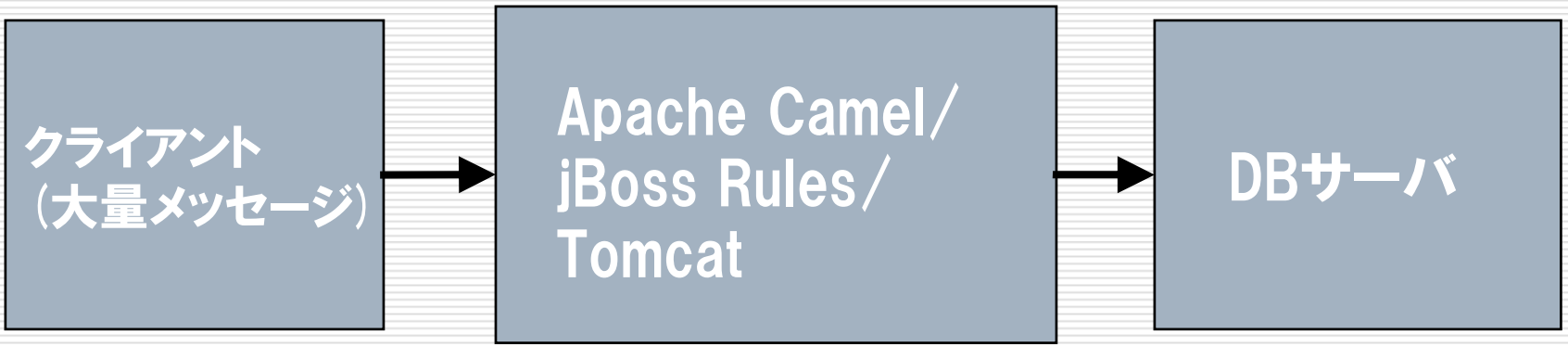
Apache Camelの適用事例紹介 ～ その① ～

- ソフトウェアスタック（EAI/BPM/APサーバ）
  - 外部システムおよび内部サブシステム接続にCamelを利用
  - 商用BPM製品のOSSリプレース（jBPMへの置換）も実施



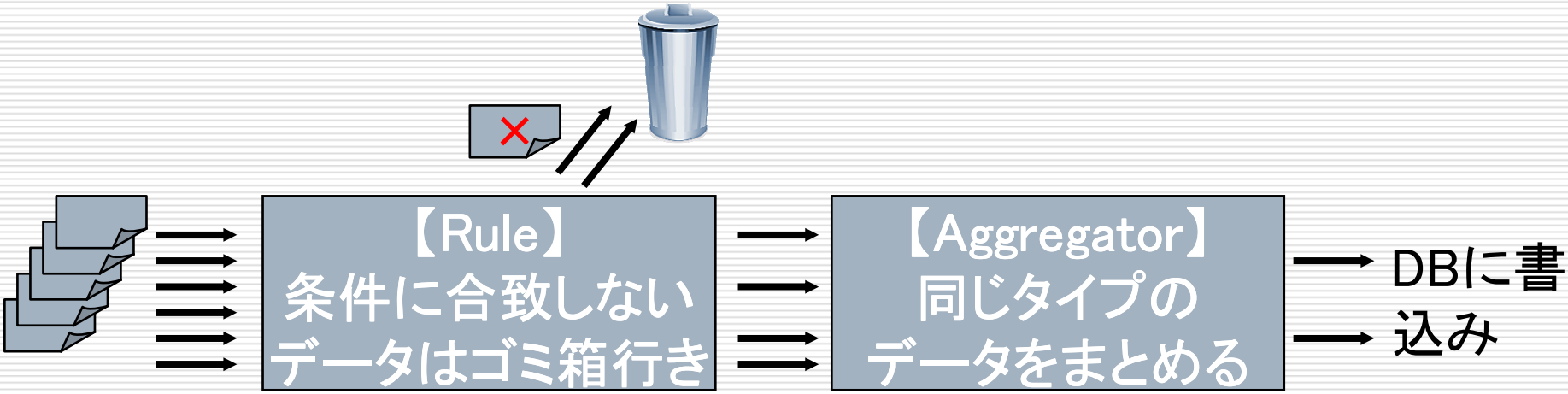
Apache Camelの適用事例紹介 ～ その② ～

- CamelとjBoss Rulesを組合せ、CEPとして利用。大量メッセージの検証。
  ※CEPとは？
  大量・多種のデータを高速で分析するもの、らしい



Apache Camelの適用事例紹介 ～ その② ～

- Ruleで大量メッセージのフィルタリング
- CamelのAggregatorでメッセージ集約
  →同類のメッセージ一つにまとめる



Apache Camelの適用事例紹介 ～ その② ～

検証してみて

- CamelのAggregator部品
  - 色々な条件でデータの集約が可能
    →時間単位、件数単位、データの中身から、等々
    →コードを書いて実現するとなると、結構大変かも
- jBoss Rules
  - Rule用の部品がCamelにはない！
    →自作するノウハウが必要

- Apache Camelとは？
- Apache Camelの適用事例紹介
- Camel概要
- Camelの使い方
- さらに深くCamelを知る
- Camelコードサンプル

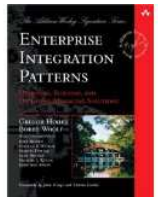
Camel概要
～特徴～

- Enterprise Integration Pattern (EIP)
- 優れた拡張性
- 多数のコンポーネント

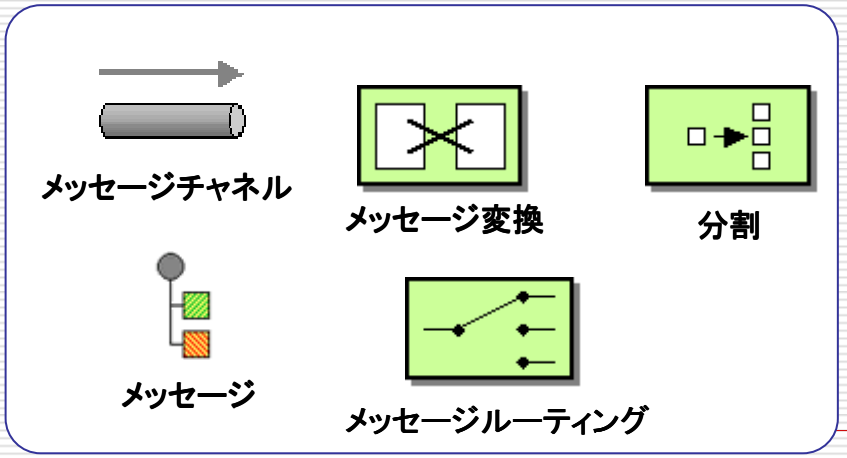
Camel概要
～Enterprise Integration Pattern (EIP)～

汎用的な設計パターンをGoFデザインパターンとして定義したように、エンタープライズ統合のパターンを定義

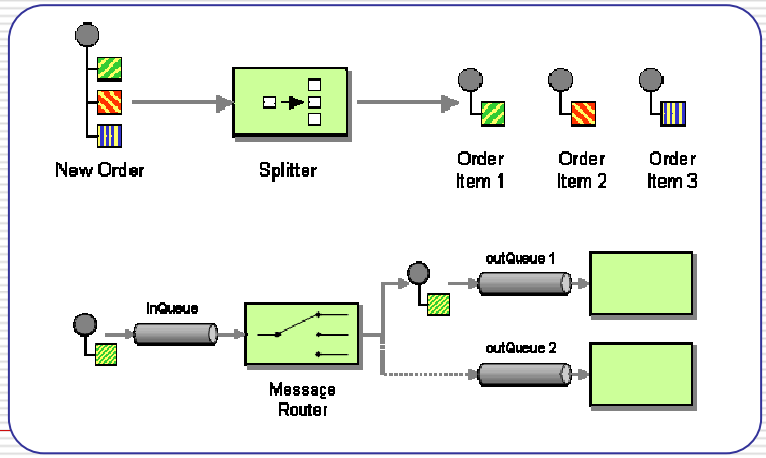
エンタープライズ統合のノウハウ → 汎用的なパターン



必要な機能を定義



利用時の留意点を定義

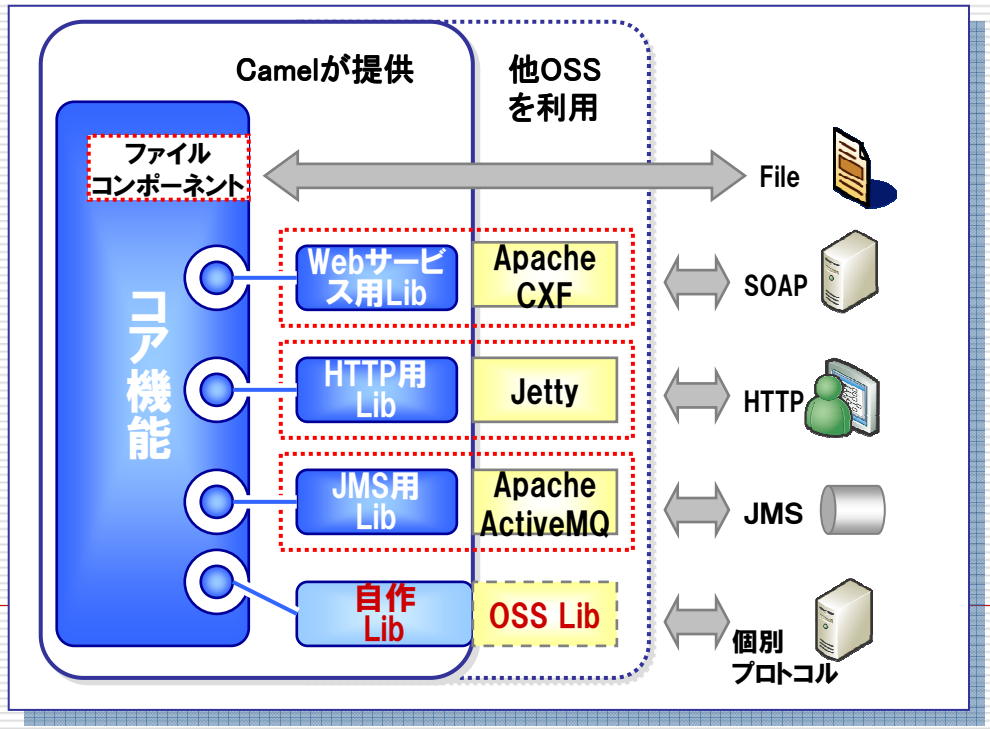


⇒CamelはEIPのパターンを参考に機能実現

Camel概要 ～優れた拡張性 1/2～

□ 必要なコンポーネントのみ利用可能

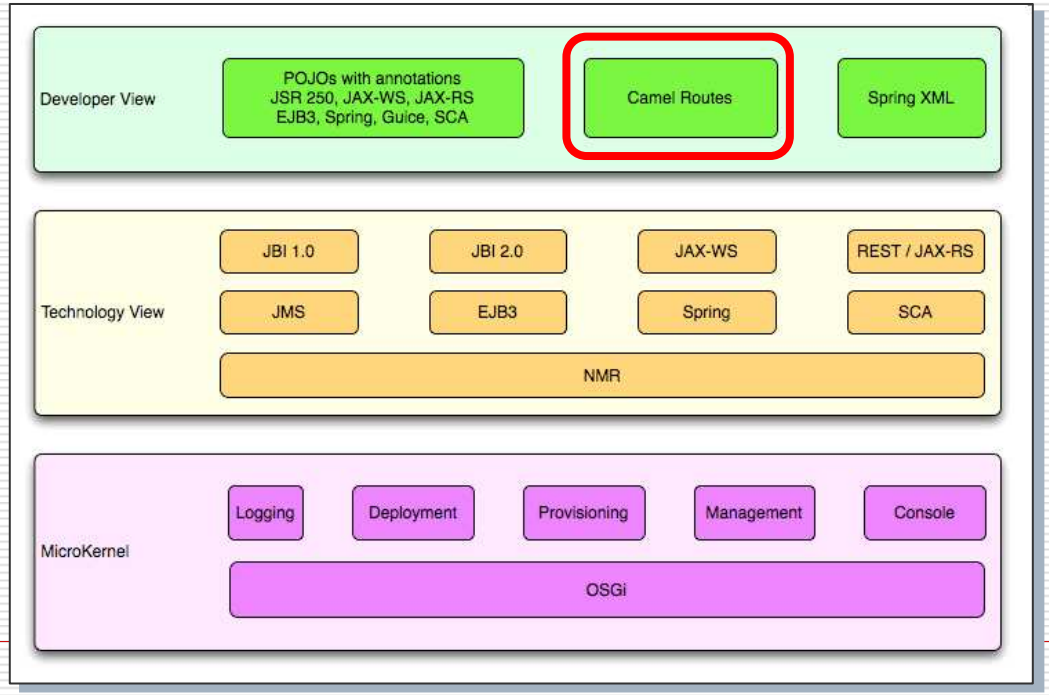
- ■ 不要なコンポーネントは使わなくてよい
- ■ ない場合は自分で作成可能



Camel概要
～優れた拡張性 2/2 ～

- **単純なプロトコルスタック**
  - **Camel本体は他ライブラリの依存度が低い**
  - **他システムへの組込み/組合せが容易**

オープンソースESBの ServiceMixの場合

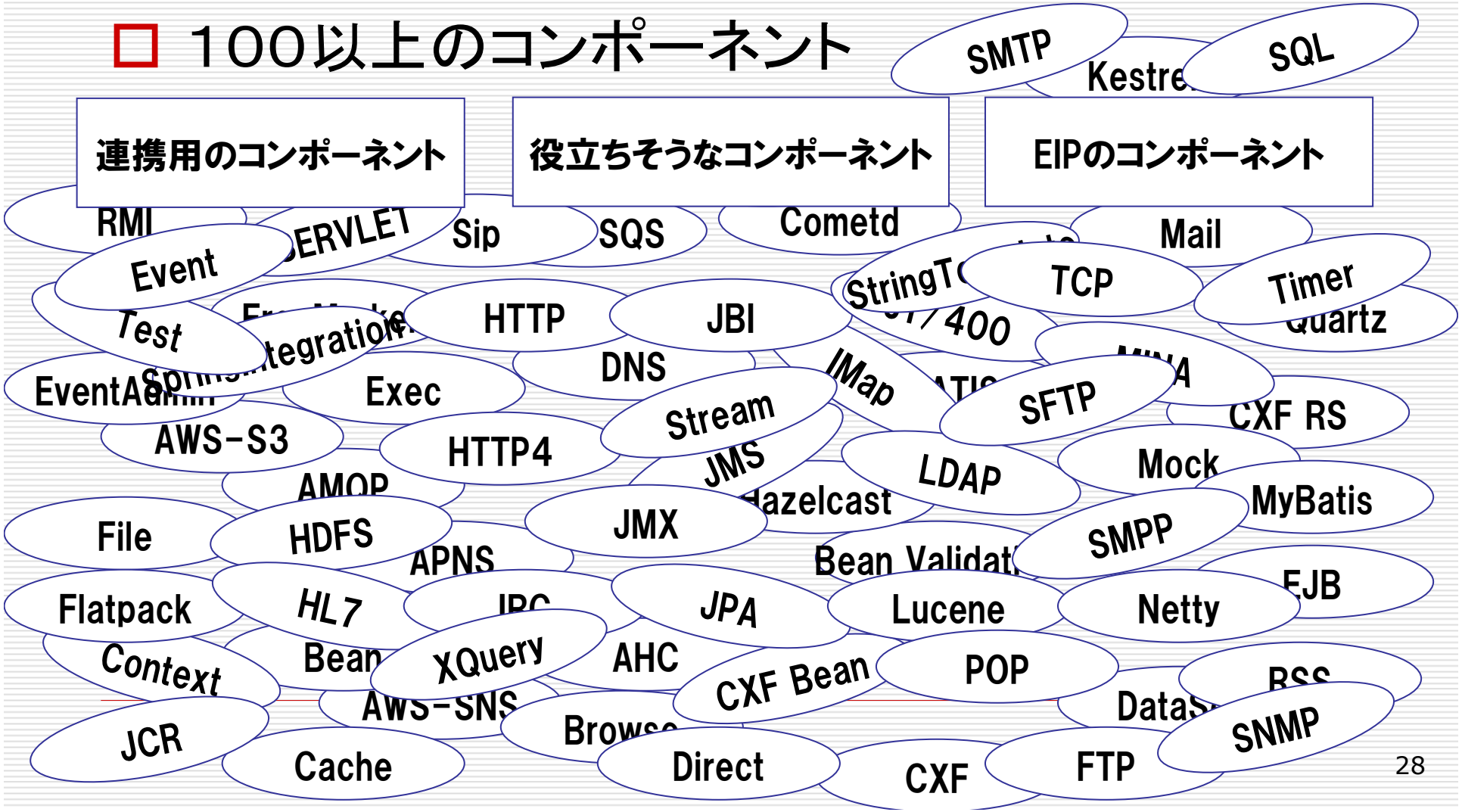


Camelと他OSSを
組み合わせて実現

http://servicemix.apache.org/architecture.htmlより引用

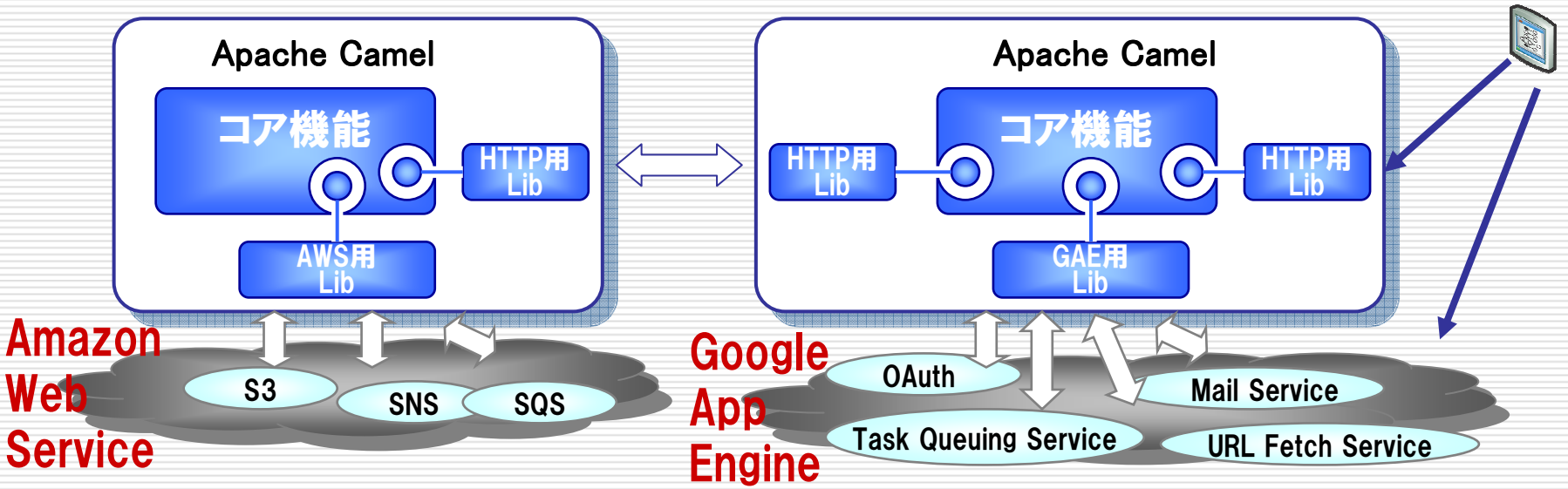
Camel概要
～多数のコンポーネント　1/2～

- 100以上のコンポーネント



Camel概要
～多数のコンポーネント 2/2～

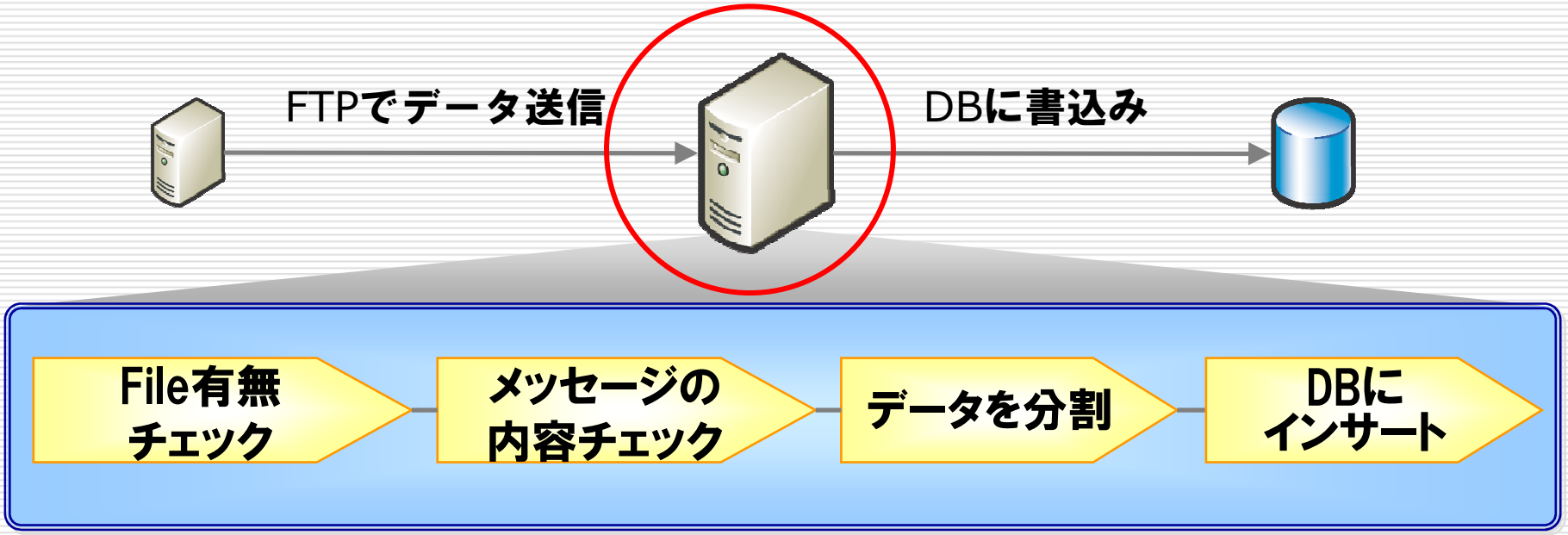
- AWS,GAE等、注目のクラウド環境にも対応



- Apache Camelとは？
- Apache Camelの適用事例紹介
- Camel概要
- Camelの使い方
- さらに深くCamelを知る
- Camelコードサンプル

Camelの使い方

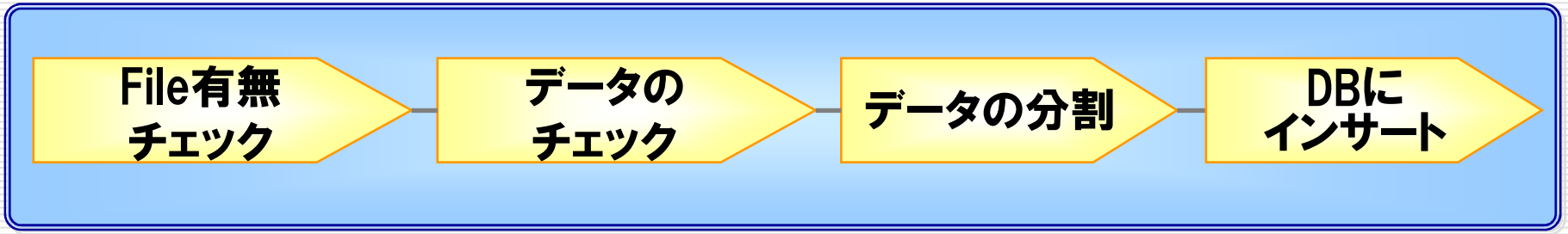
例えばこんなことしたい場合…



APサーバ上にサーブレットを作ったり、シェルを組み合わせたり、.NETやRuby等の各種言語を使ったりして実現

Camelの使い方

Camelの場合だと…



こんな風に書いて実現できます

※あくまでも雰囲気です、雰囲気。

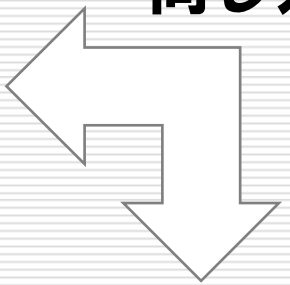
```
from("File有無チェック")
  .to("データのチェック")
  .to("データの分割")
  .to("DBにインサート");
```

Camelの使い方

XMLを使って書くこともできます。

```
<route>
  <from uri="File有無チェック" />
    <to uri="データのチェック" />
    <to uri="データの分割" />
    <to uri="DBにインサート" />
</route>
```

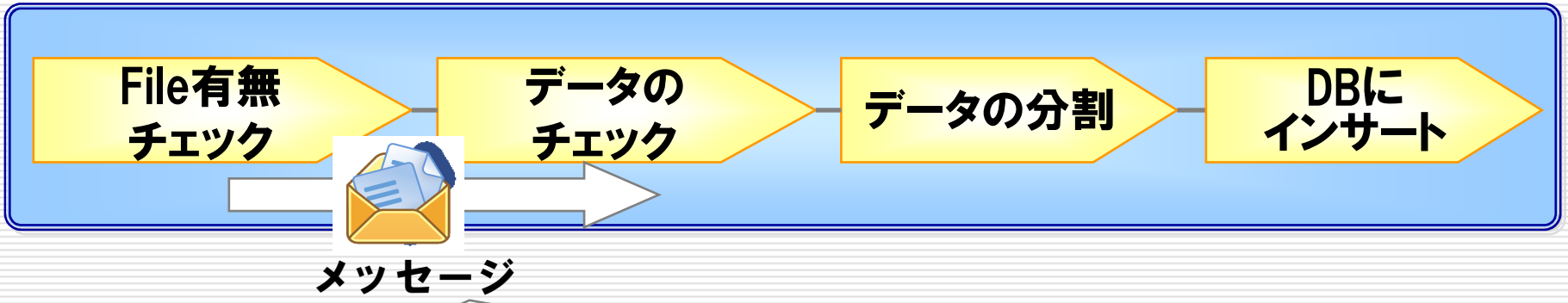
同じ処理



```
from("File有無チェック")
  .to("データのチェック")
  .to("データの分割")
  .to("DBにインサート");
```

Camelの使い方

- 実行時、Camelは記述されたルートに従ってメッセージをルーティングします。



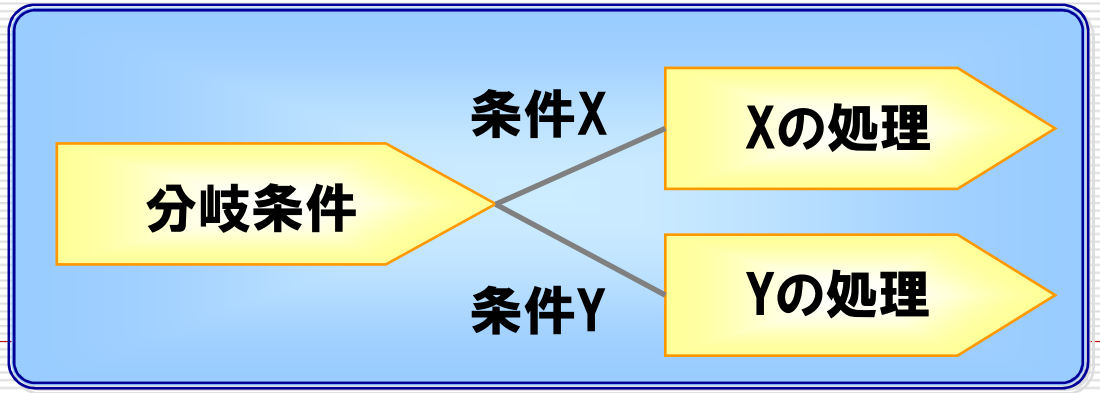
中には取得したファイルの中身は勿論、ファイル名やタイムスタンプ、取得元ディレクトリ等の情報が詰まっている

Camelの使い方 ～ルーティング～

- DSLでルーティングを定義
  - DSL：Domain Specific Language
    Camel固有のルーティング記述用の言語
- 複数のDSLでルーティング定義が可能
  - Java DSL（実際はfromやtoのメソッド）
  - Spring DSL（実際はfromやtoのタグ）
  - Scala DSL（勉強してないのでわかりません）

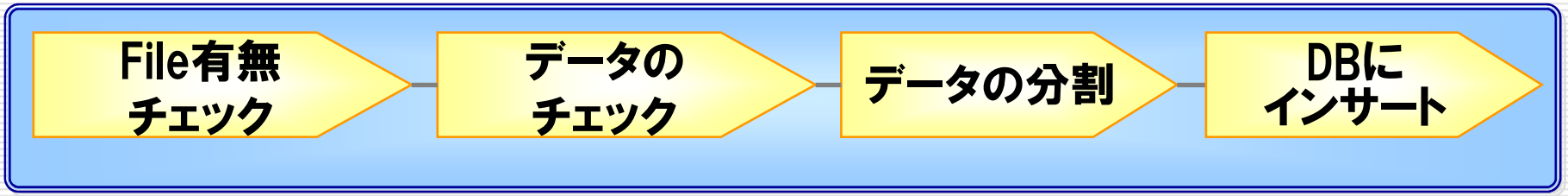
Camelの使い方 ～ルーティング～

- 処理のルートを記述するとCamelが解釈 →ルーティングエンジン
  - 基本はfromで始めてtoを続けて書く
    - Camelが解釈して順番に処理をする
  - from,to以外にも色々記述できます
  - 一直線でなく分岐させることも可能
    - choiceやwhen,otherwiseの利用



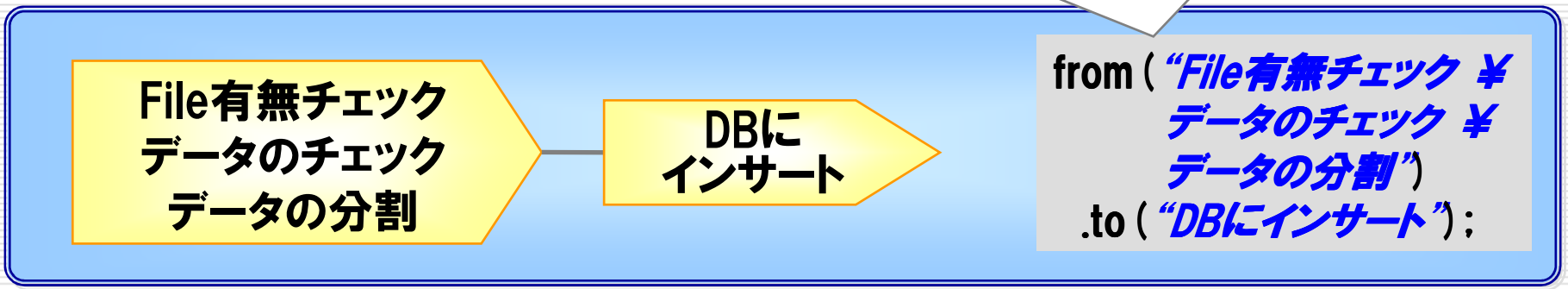
Camelの使い方

ところで、何故下記のように分けているのか？



こんな分け方は駄目なの？

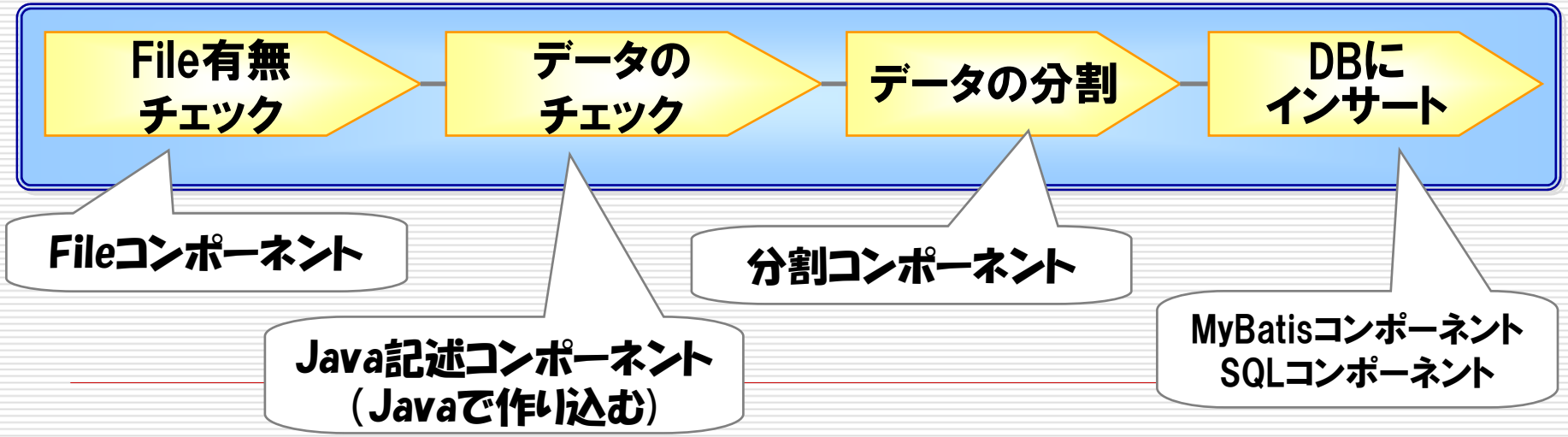
“to”の数を少なくできる？




```
from("File有無チェック ¥
      データのチェック ¥
      データの分割")
  .to("DBにインサート");
```

Camelの使い方

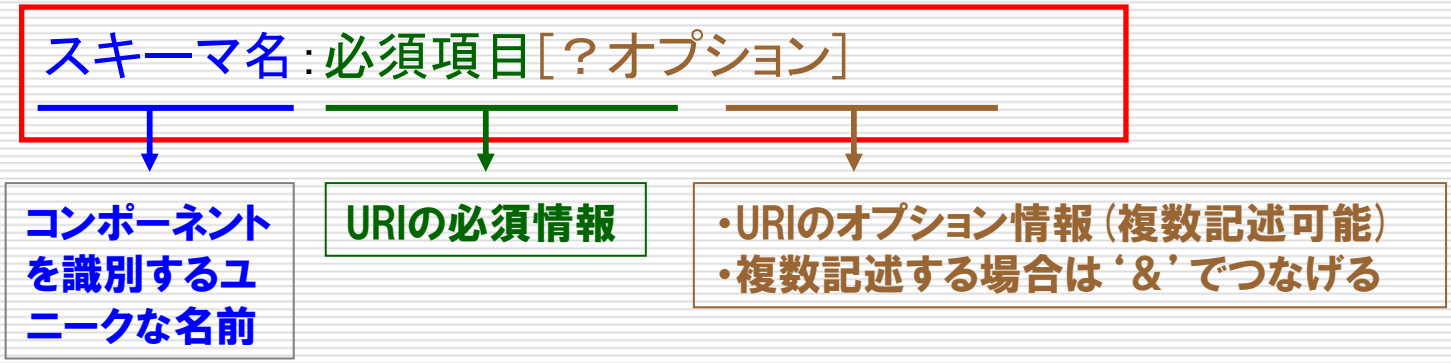
Camelで提供しているコンポーネントのサイズ



Camelの使い方 ～コンポーネント～

コンポーネントはURIで指定

〈URIの基本〉



例えば、
/tmp/abcフォルダを5秒間隔でポーリングする場合、
Fileコンポーネントを利用。スキーマ名は”file”なので、

file:/tmp/abc?delay=5000　と書ける

Camelの使い方 ～コンポーネント～

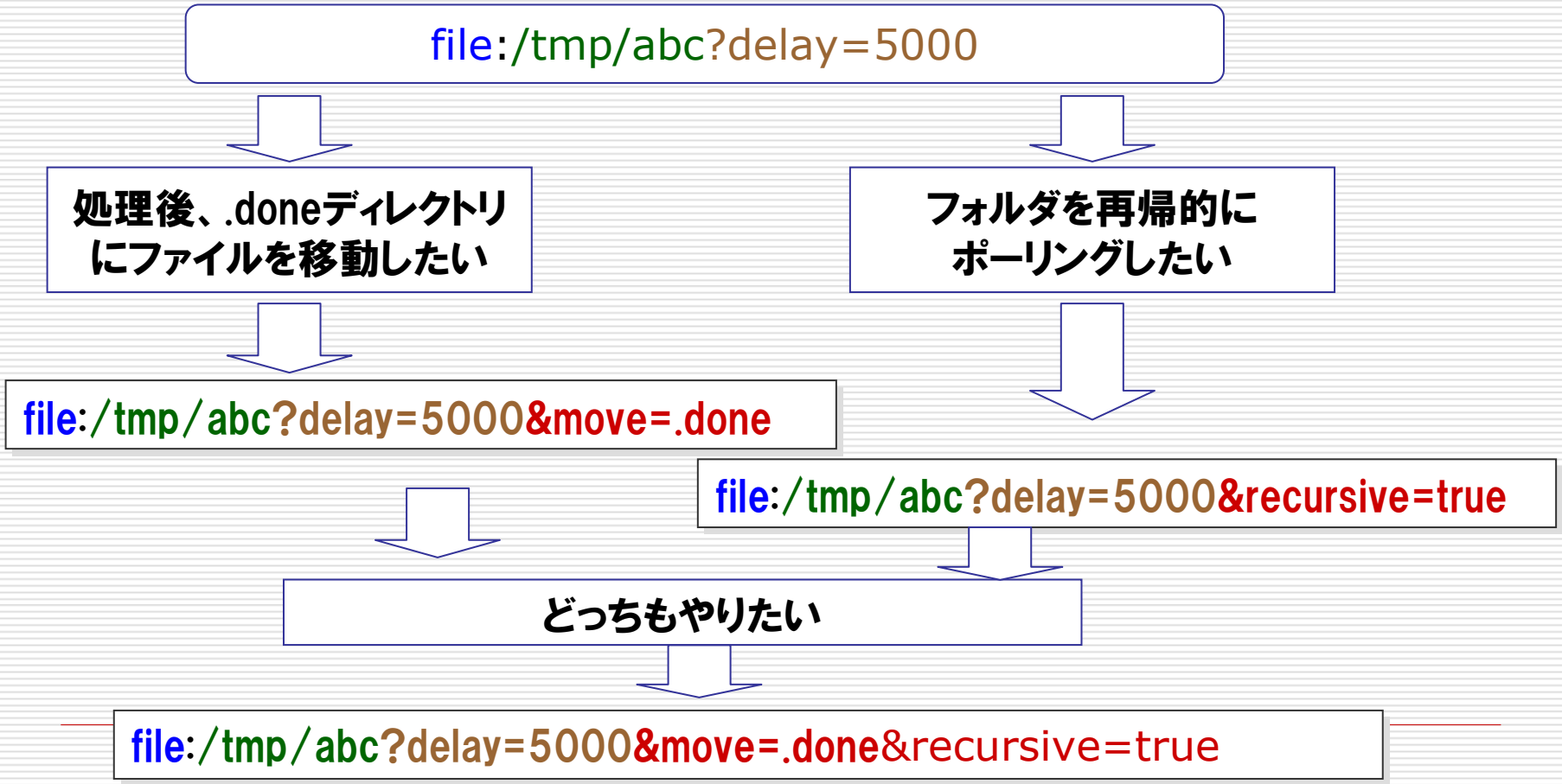
- 各種プロトコルに対応したコンポーネントを用意
  - 書式はすべて「スキーマ名：必須項目？オプション」

プロトコル	利用できるコンポーネント
File	File
Webサービス関連	CXF、CXFRS、Restlet
FTP関連	FTP、SFTP、FTPS
DBアクセス	JDBC、SQL、iBatis、MyBatis
HTTP関連	Jetty、HTTP、HTTP4
JMS、メッセージング関連	JMS、SJMS、MQTT
Amazon連携	AWS-S3、AWS-DDB、AWS-SES
Google連携	GAuth、GHttp、GLogin、GMail

他、多数。詳細は以下を参照。
http://camel.apache.org/components.html

Camelの使い方 ～コンポーネント～

- オプション指定で更に便利に！



Camelの使い方 ～ルーティング例～

- コンポーネント記述の書き方の一例


```
from(“file:/a/b?delay=5000”)
  .to(“bean:dataCheck”)
  .to(“log:com.mycompany.order?level=DEBUG”)
  .split( body().tokenize(“¥n”)
  .to( “mybatis:insertData?statementType=insert”);
```

**基本は
つなげていく！**

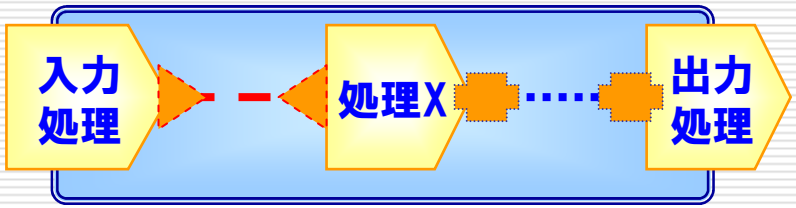
from,to**意外にも目的に応じて様々な記述が可能**

- Apache Camelとは？
- Apache Camelの適用事例紹介
- Camel概要
- Camelの使い方
- さらに深くCamelを知る
- Camelコードサンプル

さらに深くCamelを知る
～内部で流通するメッセージ形式～

- メッセージの話をする前に、少しだけ一般論

厳密に型を定義

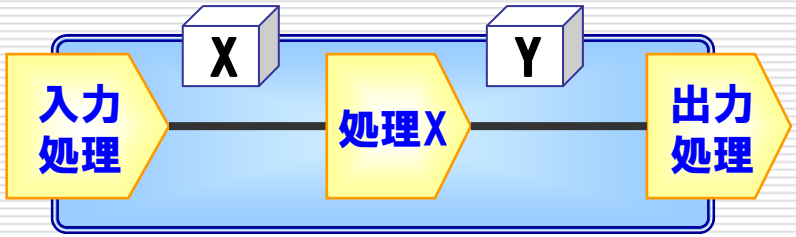


の型のみ
流通を許容

の型のみ
流通を許容

- ユーザはデータへのアクセスが容易
- 全ての型定義を設計時に実施

利用者が型の内容を意識



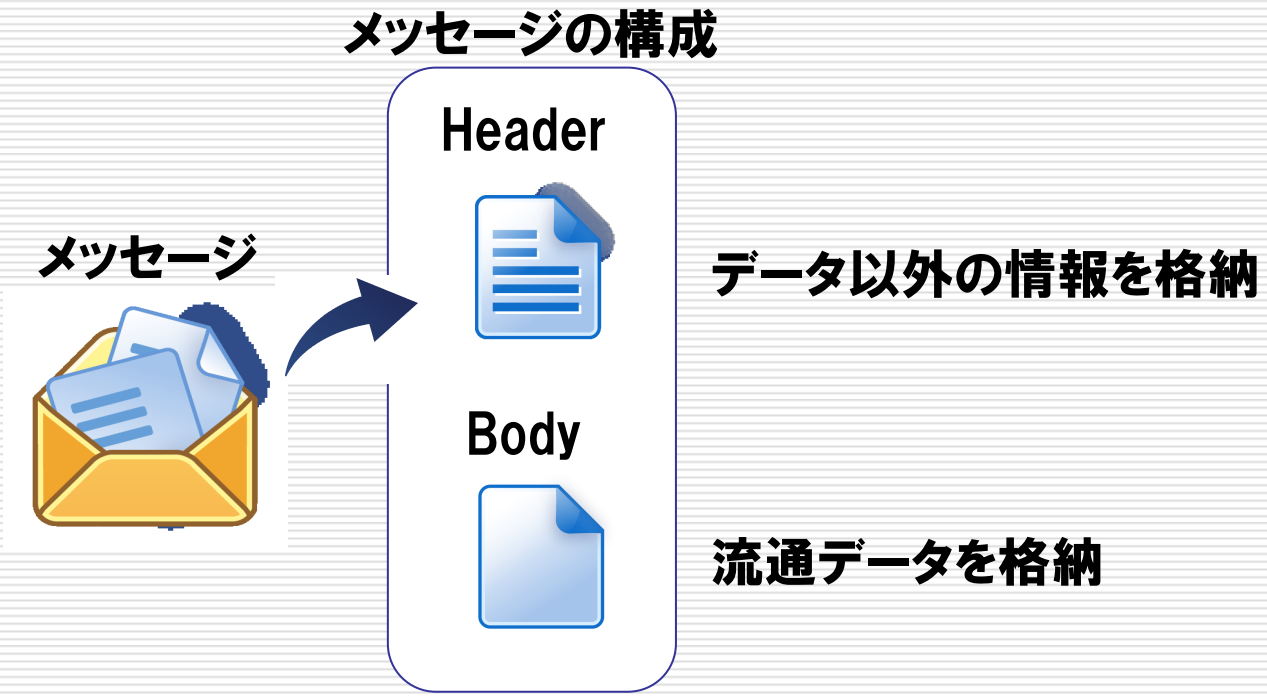
の中に何がどのように入っているか、ユーザが意識

- 設計時に型定義が不要
- データへのアクセスを各処理で実施

Camelはこの方法

さらに深くCamelを知る ～メッセージ形式～

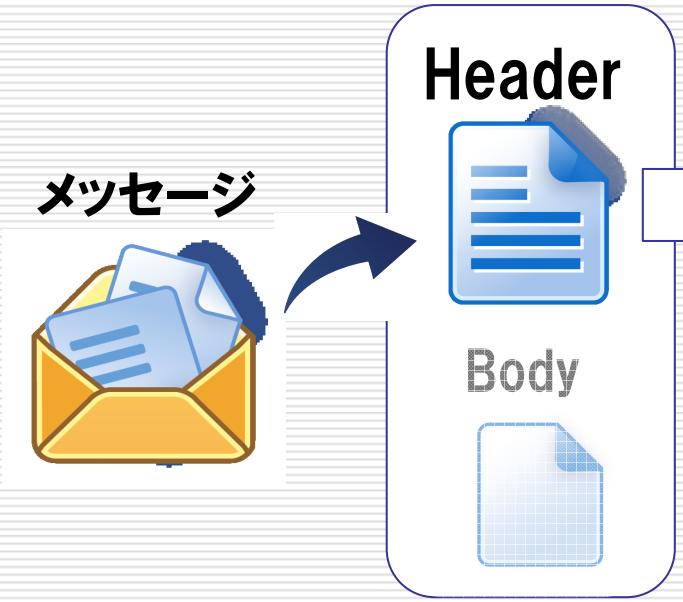
□ メッセージは“Header”と“Body”からなる



※：メッセージの包む箱としてExchangeがあるけど、今回は省略。
※：またExchange内に2つのメッセージがある（InMessage、OutMessage）

さらに深くCamelを知る ～メッセージ形式　Header～

- データ以外の情報がHeader部に詰まっている



- ◆例えば、ファイル名やディレクトリ名を格納
- ◆「キー＝値」の形式で格納
- ◆プロトコル毎に格納されるキーが異なる
- ◆ユーザが自由に追加/変更可能

ファイル受信時のヘッダ情報の例

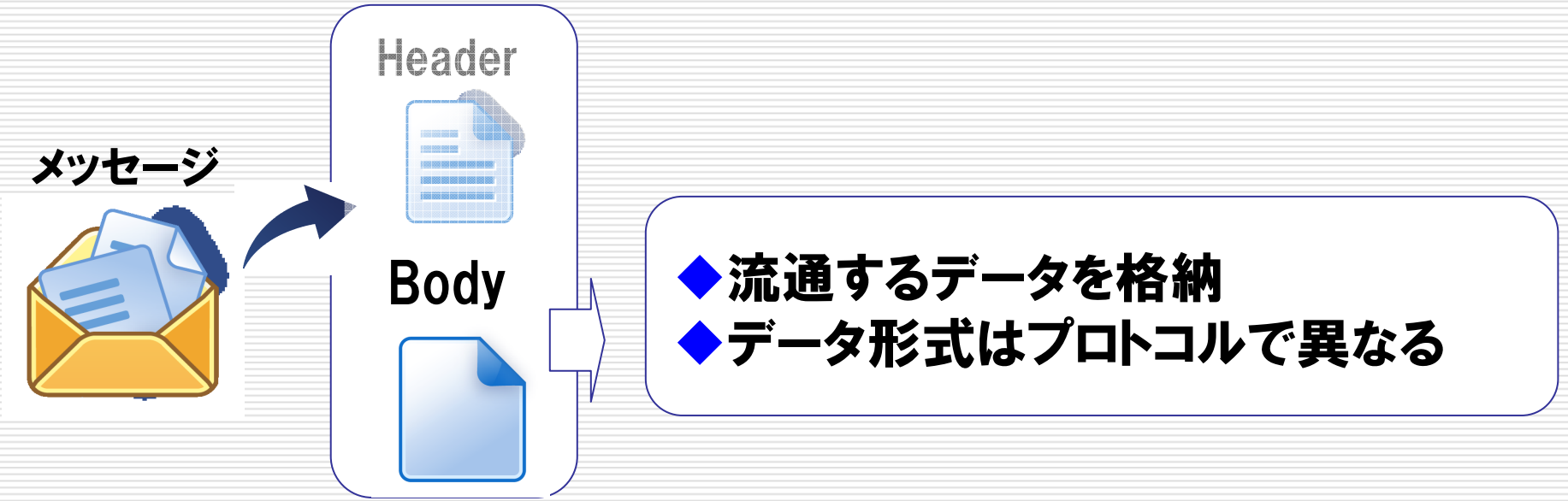
```
[CamelFilePath=C:¥work¥abc¥File.txt]
[CamelFileLength=320]
[CamelFileLastModified=yyyy/mm/dd]
…
```

Webサービス受信時のヘッダ情報の例

```
[Content-Type=text/xml;charset=UTF-8]
[operationNameSpace=http://ws.test.co.jp/]
[operationName=helloEcho]
…
```

さらに深くCamelを知る ～メッセージ形式 Body～

- Body部には流通データが詰まっている



ファイル受信時のBody部

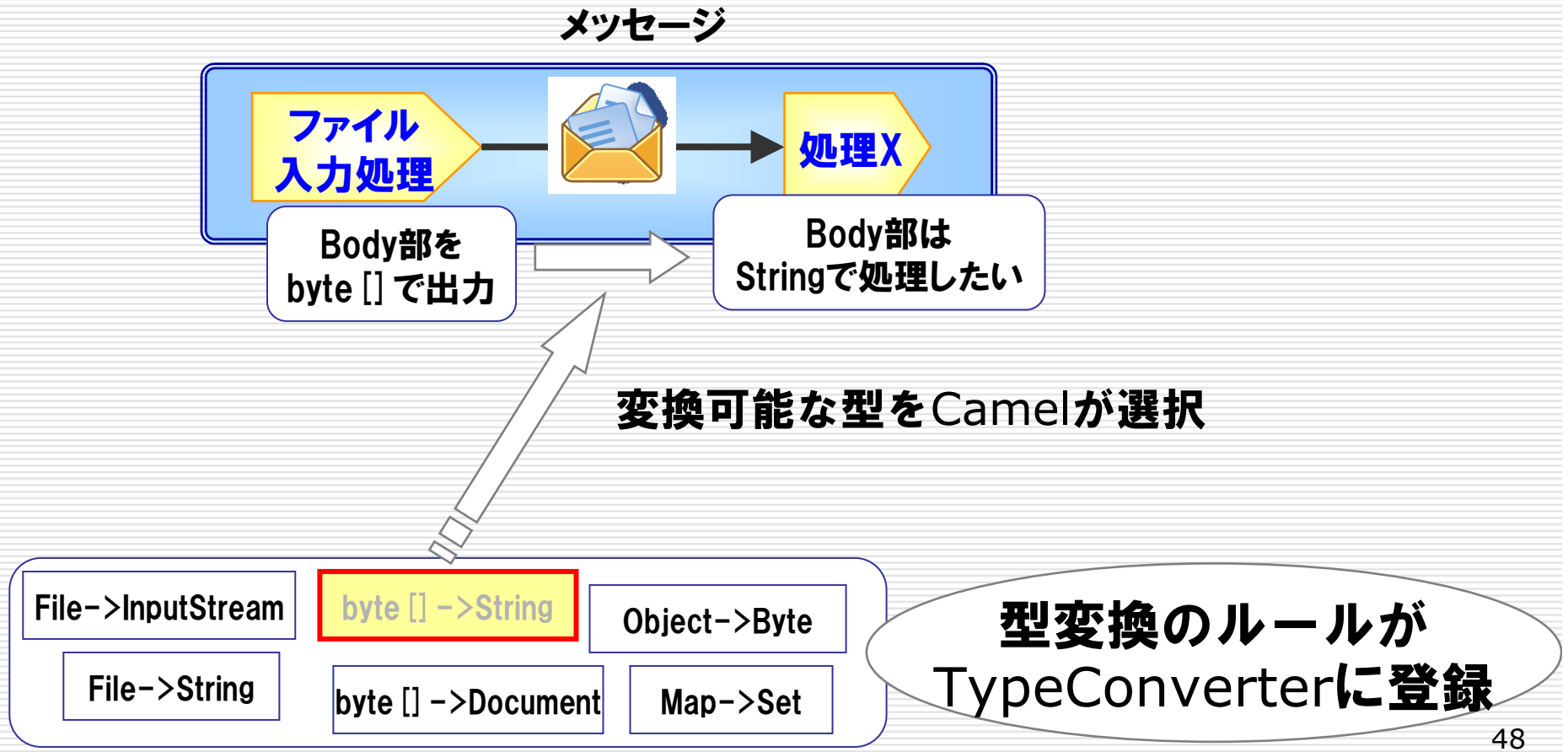
◆クラス名は
org.apache.camel.component.file.GenericFile
◆処理対象のファイル情報を格納

Webサービス受信時のBody部

◆クラス名は
org.apache.cxf.message.MessageContentsList
◆受信データのリスト、引数のデータが順番に格納

さらに深くCamelを知る ～データ変換～

- 自動でBody部の型変換をするTypeConverter



さらに深くCamelを知る ～データ変換～

□ 容易にBody部のデータ取得が可能

「処理X」はいろんな型でデータが取得できる

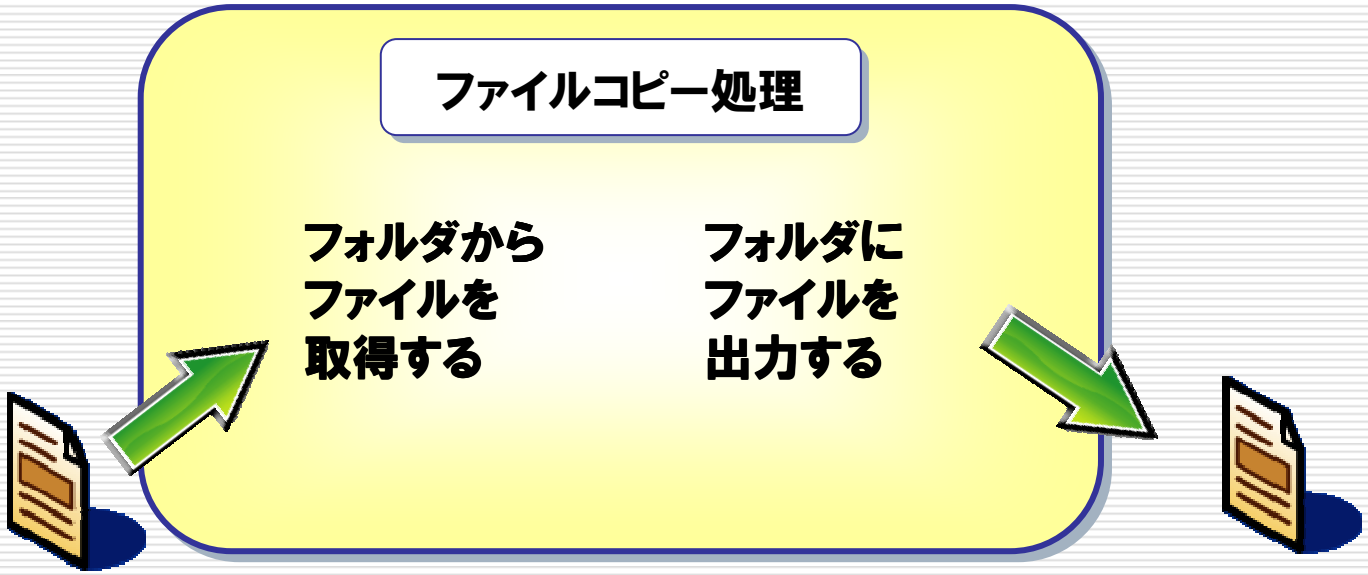
Body部をXXXの型でくれ！
と利用者が宣言するだけ

```
public class Process implements Processor {
  public void process(Exchange exchange) throws Exception {

    //Stringで欲しい場合
    String data = exchange.getIn().getBody(String.class);

    //InputStreamで欲しい場合
    InputStream is = exchange.getIn(). getBody(InputStream.class);

    //Documentで欲しい場合
    Document doc = exchange.getIn(). getBody(Document.class);
```

- Apache Camelとは？
- Apache Camelの適用事例紹介
- Camel概要
- Camelの使い方
- さらに深くCamelを知る
- Camelコードサンプル

Camelコードサンプル ～実際にやってみる～

- 例えば…
  ファイルをコピーする処理を考えてみる



Camelコードサンプル ～Javaで書いてみる～

Javaでも簡単に記述可能

```
public static void main(String args[]) throws Exception {
  File inboxDirectory = new File("data/inbox");
  File outboxDirectory = new File("data/outbox");
  outboxDirectory.mkdir();
  File[] files = inboxDirectory.listFiles();
  for (File source : files) {
    File dest = new File(outboxDirectory.getPath() + File.separator + source.getName());
    copyFile(source, dest);
  }
}

private static void copyFile(File source, File dest) throws IOException {
    ～ファイルのコピー処理をする～
}
```

Camelコードサンプル ～Camelでやってみる～

Camelの場合(Java DSL)

```
public static void main(String args[]) throws Exception {

  //1:最初にCamelContextを作成する
  CamelContext context = new DefaultCamelContext();
  RouteBuilder routeBuilder = new FileToFileRoute();
  context.addRoutes(routeBuilder);

  //2:作成したCamelContextを開始し、10秒後に終了する
  context.start();
  Thread.sleep(10000);
  context.stop();
}
```

```
public class FileToFileRoute extends RouteBuilder {

  @Override
  public void configure() throws Exception {
  from("file:data/inbox?noop=true")
        .to("file:data/outbox");
}
```

Camelコードサンプル ～要件の考慮～

- ところが、普通はもっと要件が複雑…
  例えば、
  - 1分毎にファイルがあるかチェックすること
  - 出力ファイル名には日付を付与すること
  - .docの付いた拡張子は無視すること
  - 同一のファイル名がある場合は無視すること
  - サブフォルダも検索すること
  - コピーでなく移動に変更。バックアップも取って！
  - etc,…

Camelコードサンプル ～追加要件への対応～

- Camelの場合、オプション追加で対応可能

1分毎にファイルがあるかチェック

.docの付いた拡張子は無視

.doneフォルダにバックアップ

```
public class FileToFileRoute extends RouteBuilder {

  @Override
  public void configure() throws Exception {
  from("file:data/inbox? delay=60000 & exclude=*.doc$ & move=.done &
                         noop=false & idempotent=true & recursive=true")
        .to("file:data/outbox? fileName=${date:now:yyyy-MM-dd}_${file:name}");
}
```

サブフォルダも検索

コピーから移動に変更

出力ファイル名には日付を付与

同一のファイル名がある場合は無視

さいごに(その①)
～様々な立場におけるApache Camel～

□ インテグレータ View

- ■ 安価なシステム提案

  → イニシャルコストなし

  → ランニングコストはメンテナンスに必要な稼働を請求！？

- ■ 国内でApache Camel先駆者となれる

  → 国内での利用ユーザはほんの僅か、
  使い倒して競合他社との差別化！？

- ■ 容易な開発体制の確立

  → 専門技術者を集める必要なし、Java技術者にて開発可能

さいごに(その①)
～様々な立場におけるApache Camel～

□ エンドユーザ View

- ■ 安価なシステム構築

→ イニシャルコストなし

→ 必要なのはメンテナンス費用(≒保守費)

→ ポジティブに考えるといつでも捨てれる！？

→ OSSの積極的な利用によるコストダウン体制をアピール

さいごに(その②)
～ユーザ会を立ち上げました～

- 皆さんにCamelを知って欲しい！
- 日本Apache Camelユーザ会

【お勉強系サイト】
http://sourceforge.jp/projects/cameluserjp/

【キャラ系サイト】

https://sites.google.com/site/jaacug/

- 主な活動内容
  - Camelのドキュメント翻訳
  - Camelのサンプル実装

等をやりたいと思っています。

ご静聴
ありがとうございました