

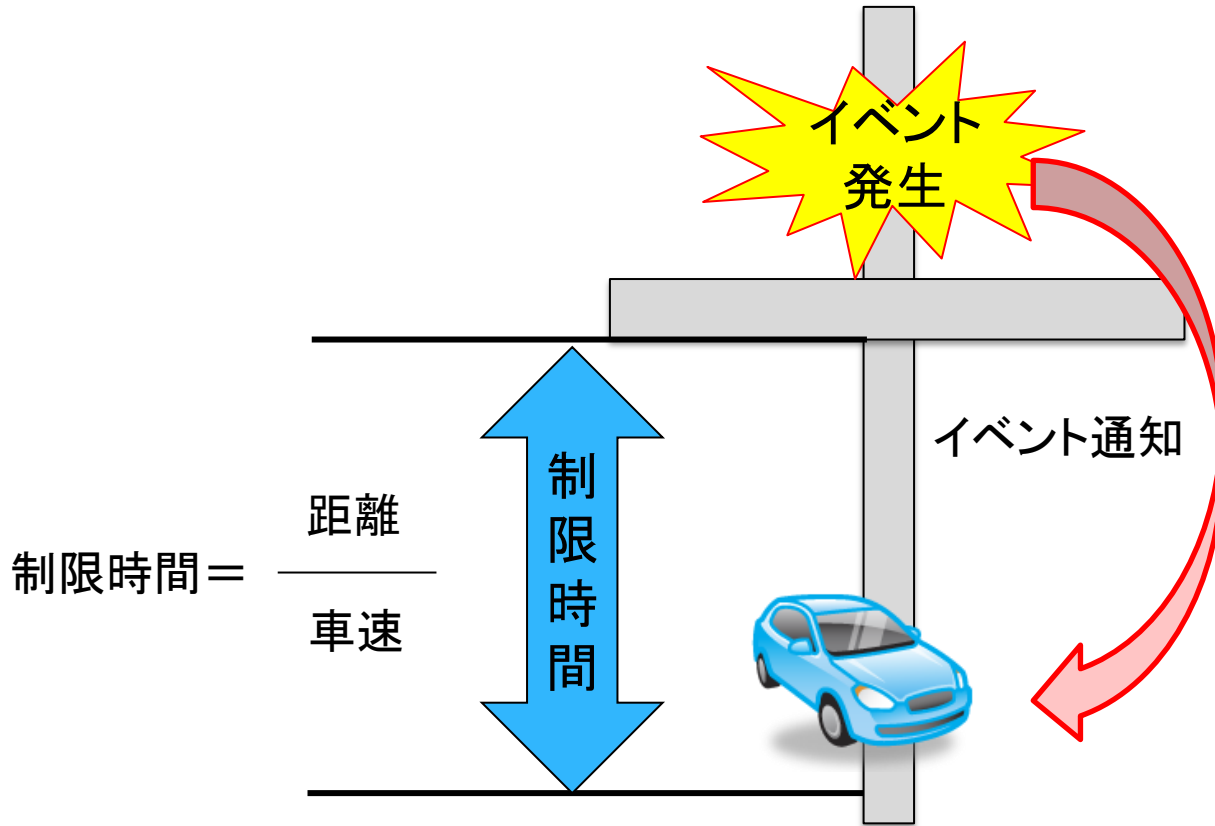
# 自動車の外部連携サービス指向 アーキテクチャの提案

南山大学大学院 数理情報研究科  
数理情報専攻 ソフトウェア工学専修  
M2011MM030 伊藤 智基  
指導教員: 青山 幹雄

- 質疑応答の回答
  - 質問1
  - 質問2
  - 質問3
- 今後の予定
- 参考文献

# 中間質疑応答の回答(1/3)

- リアルタイム性はどの**程度のレベル**を考えている？



参考までに:

ハードリアルタイム: エンジン制御(100 $\mu$ sオーダー)

ソフトリアルタイム: ルート案内(ms~sオーダー)

# 中間質疑応答の回答(1/3)

## ■ 考察

- 同時に複数のイベントが発生⇒それぞれのイベントを制限時間内に処理する必要性
- 例:カーナビゲーションシステム

イベントごとに時間制約が異なる

### カーナビゲーションシステム

時間制約が厳しい処理

GPSからの位置情報処理

車速情報処理

時間制約が緩い処理

ルート表示

時間制約が緩い処理よりも優先的に実行し、時間制約を達成

運転者に対して実行される処理

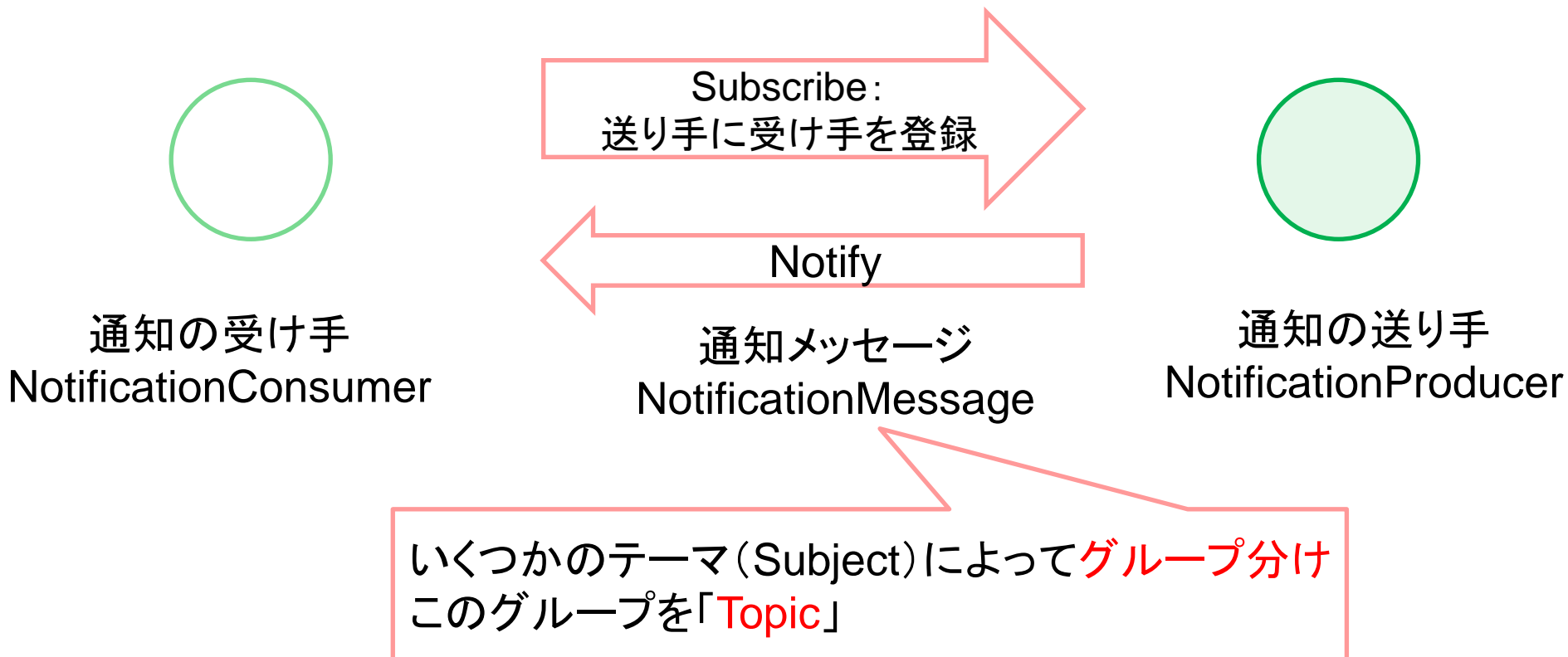
運転者は常に画面を見ることはできない。よって数秒単位で処理が完了できれば良い

- 具体的にどう技術を用いたらリアルタイムを達成できる？
  - WS-Notification: パブリッシュ/サブスクライブ型のメッセージ通信
    - WS-Based Notification: 基本的な部分
      - **Grid/WSRFとビジネスプロセスの統合“WS-Notificationの基礎”**
    - WS-Brokerd Notification: メッセージのの送り手と受け手の間に仲介役「ブローカ」の役割を規定
      - **Web Services Brokered Notification 1.3**
    - WS-Topics: メッセージを分類するTopic構造を規定

今回は  
WS-Based Notification  
WS-Brokerd Notification  
について

## ■ パブ/サブ型通信の3つの基本的なパターン

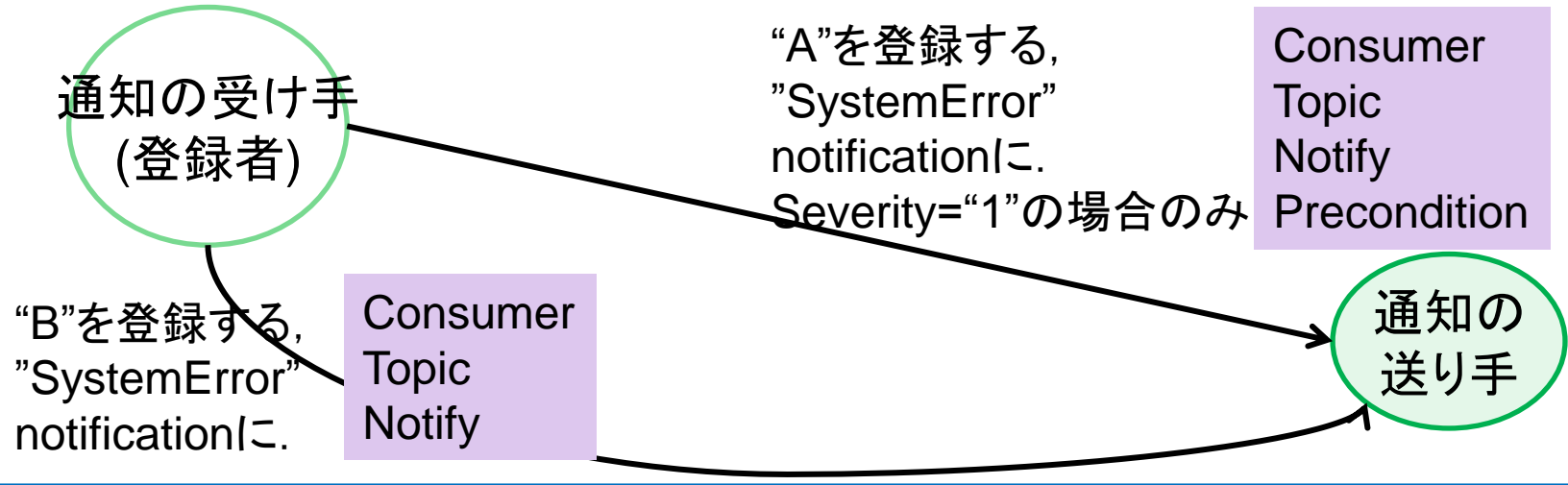
- メッセージの送り手と受け手, メッセージが存在
- 送り手側にメッセージの受け手と, 受け手の興味の対象「Topic」の登録
- 送り手のメッセージ送信は, 受け手のあるメソッドの呼び出しとして実行



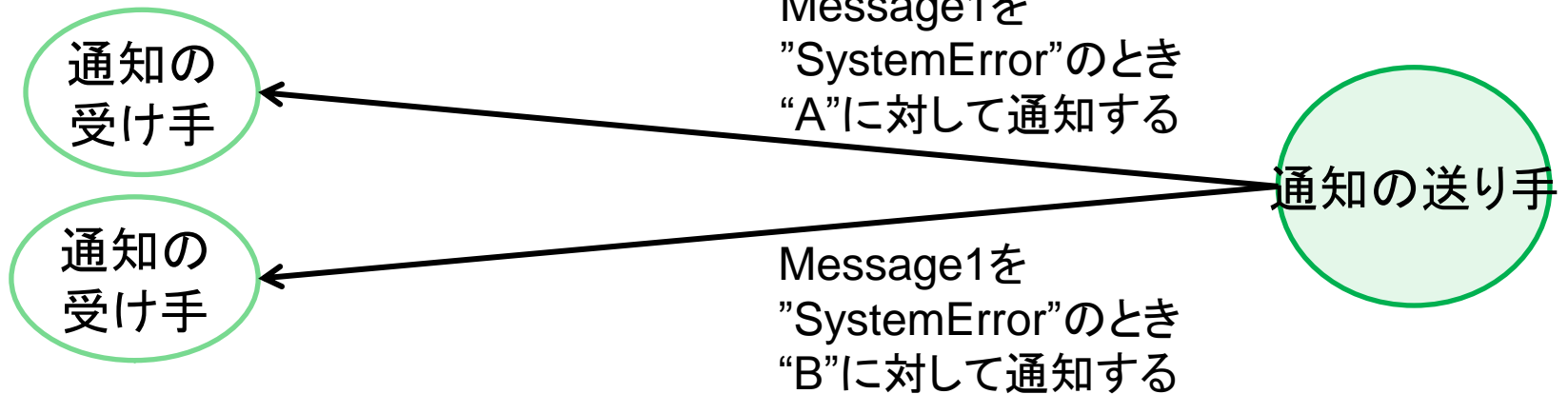
# 中間質疑応答の回答(2/3)

## ■ WS-Notification Subscribe

- “A”(Consumer)を登録する, ”SystemError”(Topic)notificationに. Severity=“1”の場合のみ(Precondition)



## ■ WS-Notification Notify



# 中間質疑応答の回答(2/3)

## NotificationProducerのWSDL定義

```

<wsdl:portType name="NotificationProducer"
  wsrp:ResourceProperties = "wsnt:NotificationProducerRP">
  .....
  <wsdl:operation name="Subscribe">
    <wsdl:input message="wsnt:SubscribeRequest"
    <wsdl:output message="wsnt:SubscribeResponse" />
  .....

```

リソースプロパティ

オペレーション定義

## NotificationConsumerのWSDL定義

```

<wsdl:portType name="NotificationConsumer">
  <wsdl:operation name="Notify">
    <wsdl:input message="wsnt:Notify" />
  </wsdl:operation>
</wsdl:portType>

```

オペレーション定義

output要素は無し



## Notificationサンプル

- サーバ上のリソース(WS-Resource)でカウンタ値が変更されたとき, クライアントにそのことをロールバックするGT4(Globus Toolkit4)のCounterサンプル
  1. ServiceAddressingLocatorのインスタンスを生成
  2. Endpointに対してget\*Portを行って, Stubを獲得
  3. Stub上のcreate\*は, サーバ上でcreate\*を呼び出す
  4. サーバ上のcreate\*はHomeクラスのcreateメソッドを通じてリソースを生成
  5. Stub上のcreate\*メソッドの呼び出しは, レスポンスを返す
  6. ResponseからEndpointReferenceを獲得

- 環境:GT4(Globus Toolkit 4)

1. J2SE:Java言語の標準的な機能をまとめたもの
2. Apache Ant:ビルドの実行を補助するツール
3. Junit:単体テストの実行を支援するフレームワーク

# 中間質疑応答の回答(2/3)

## WS-BrokeredNotification

### ■ WS-BrokeredNotification仕様での定義

#### ➤ Publisher

□ Notification の生成元となる NotificationProducer

□ Webサービスである必要なし

□ NotificationBroker に対する Notify を特に Publish と呼ぶ

#### ➤ NotificationBroker

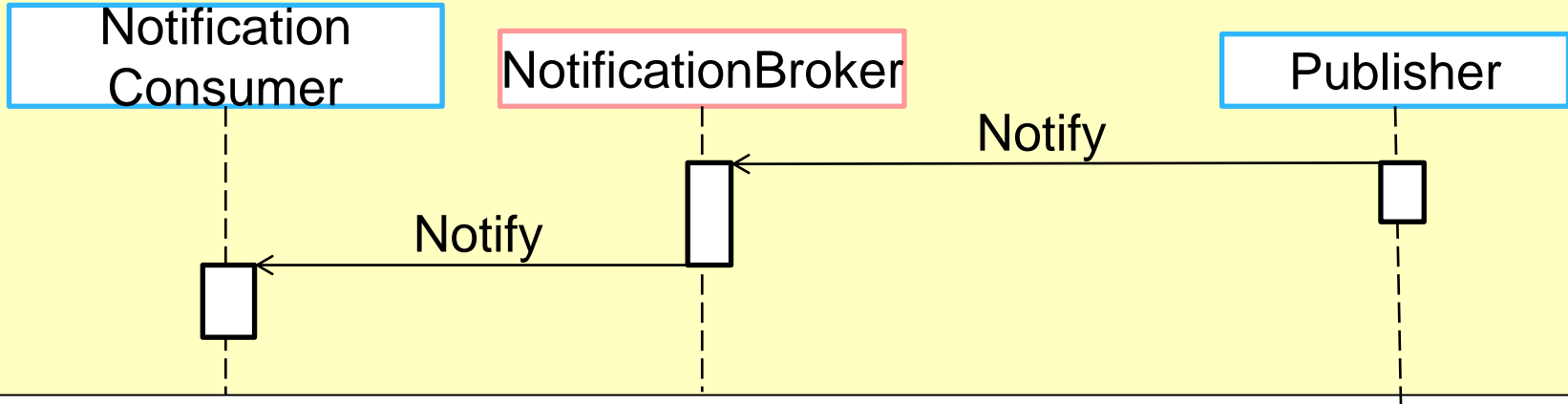
□ PublisherとNotificationConsumerを分離し, 仲介するWebサービス

- ✓ NotificationProducerにとっては, 誰にNotifyすべきかを知る必要は無い
- ✓ NotificationConsumerにとっては, NotificationBrokerに欲しい情報の取得を依頼することで, 探す手間が省ける
- ✓ 複数のNotificationMessage をマージするような機能は対象外

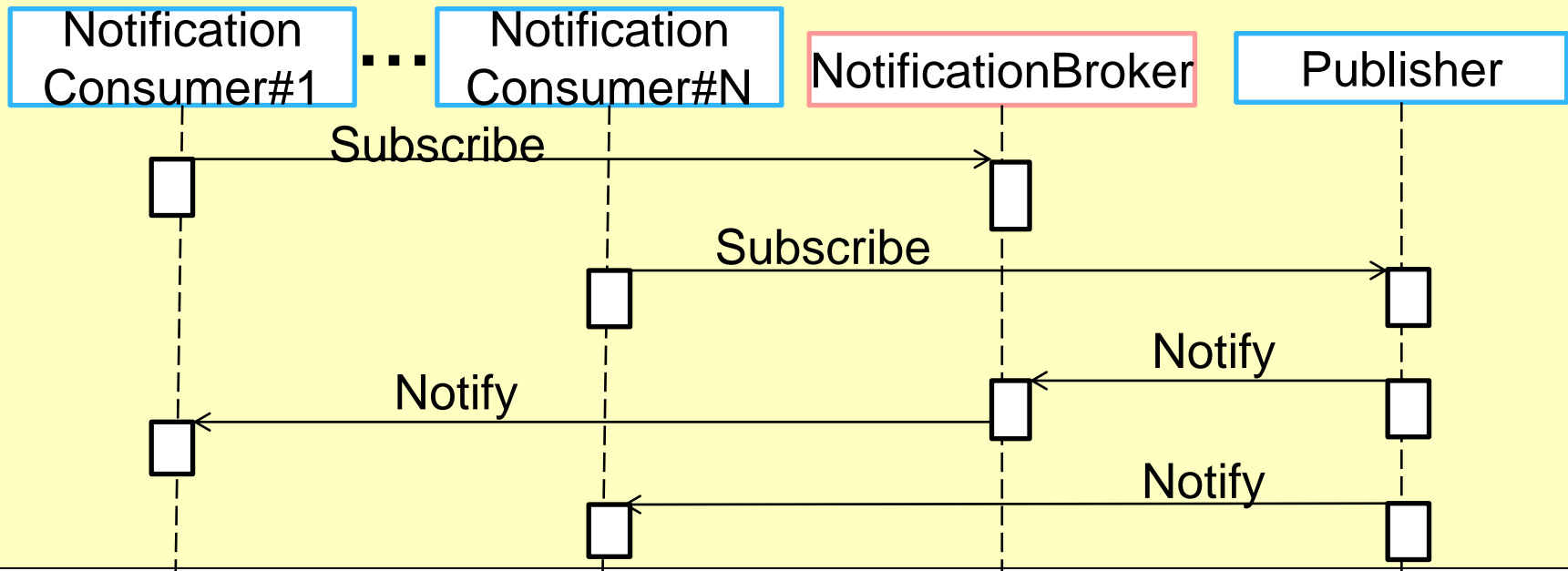
# 中間質疑応答の回答(2/3)

## WS-BrokeredNotification

simple publishing

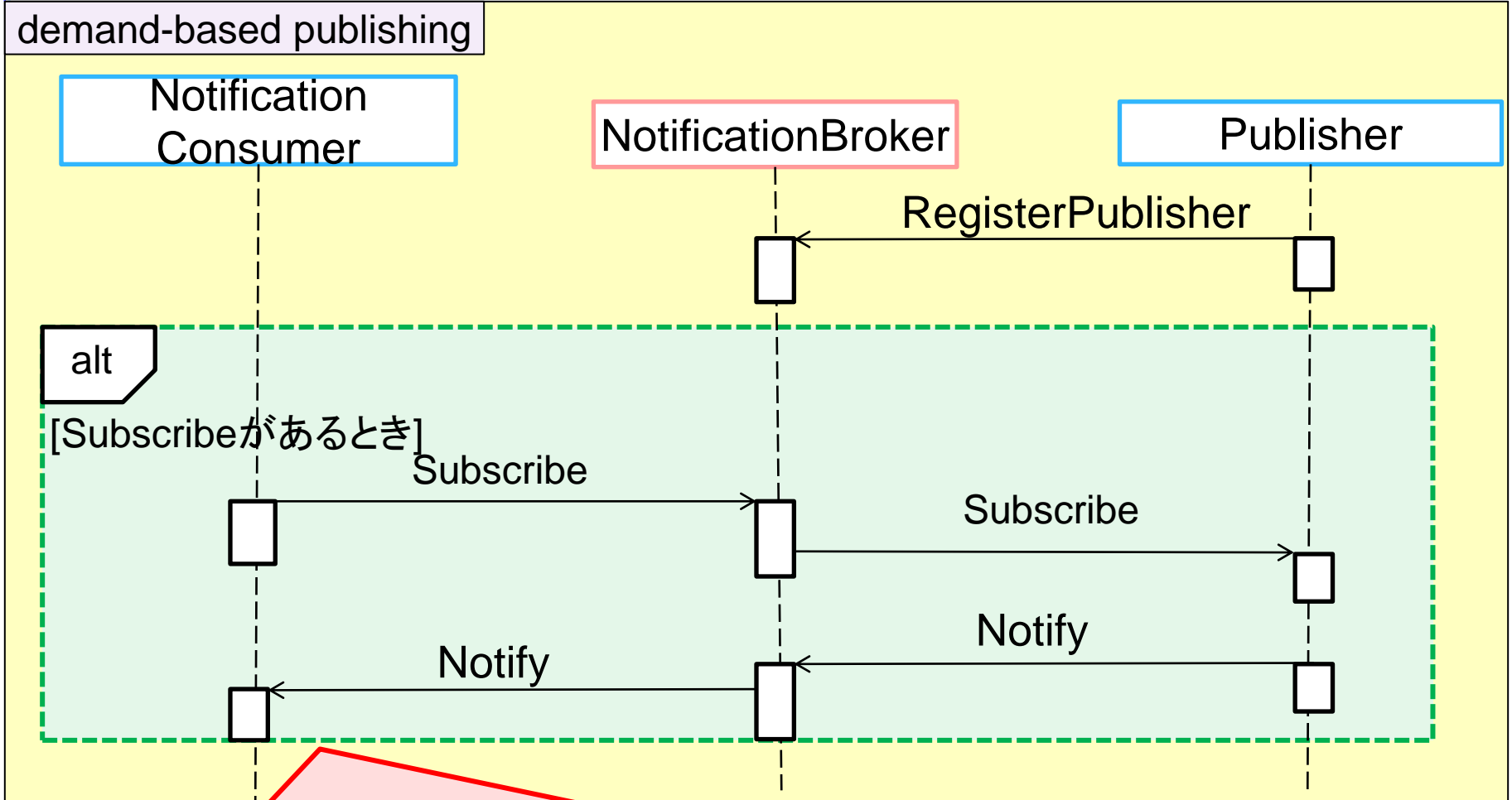


broker-initiated publishing



# 中間質疑応答の回答(2/3)

## WS-BrokeredNotification



- subscriber が現れたときに, はじめて notify 生成
- NotificationBrokerに対して, 誰も subscribe していなければ, publisherから NotificationBrokerに対して notify していても, その情報は利用されない

## ■ 考察

- 各コンポーネントの役割を基にブローカの機能を割当て⇒ブローカーアーキテクチャの具体化

## ■ リアルタイム性を考えるとき、デッドラインなどの例外処理は？

- 現在は想定していない

- Web Services Topics 1.3の読解(~10/25)
- ブローカーアーキテクチャの提案(10/26~)
- アーキテクチャの見直し(~11/12)

# 参考文献

- 組込みシステムとリアルタイムOS“2章”
- 組込みシステム概論
- グリッドとSOAからみるWebサービス標準技術“WS-Notificationの基礎”
- Grid/WSRFとビジネスプロセスの統合“WS-Notificationの基礎”
- Web Service Brokered Notification 1.3

# 自動車の外部連携サービス指向 アーキテクチャの提案

END

南山大学大学院 数理情報研究科  
数理情報専攻 ソフトウェア工学専修  
M2011MM030 伊藤 智基  
指導教員: 青山 幹雄