

NGN/IMS時代のWebアプリケーション開発

クライアント間マッシュアップ用UA Servletアプリケーションサーバ
「雷電」の紹介とデモ

SIPpropプロジェクト



自己紹介

- 氏名：今村謙之（いまむらのりつな）
- 年齢：29歳＋15ヶ月
- IT業界歴：約7年
- SIP歴：約5年
- 得意言語：Java、C
- 得意分野：SIP、ネットワーク層
（開発～運用、セキュリティー）

- 特記事項：PCサーバ タワー8台運営中(自宅にて)

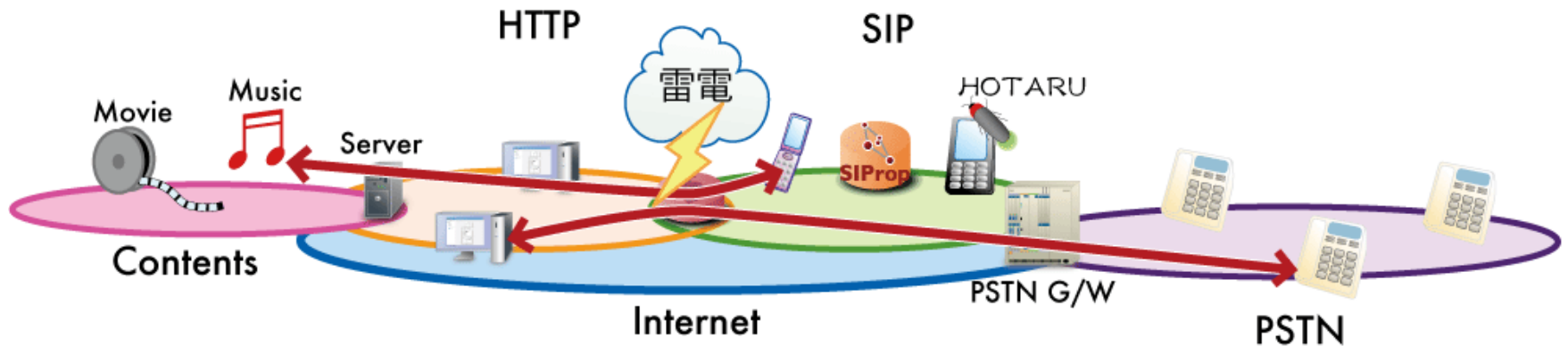
- SIPPropプロジェクトとは
 - 各種プロジェクト概要

- 雷電
 - 背景
 - 設計
 - デモ
 - 可能性

SIPropプロジェクト解説編

「メディア（通信媒体）の世界を広げる」

という使命の元、SIPの汎用セッションプロトコルという
特性に注目し、この特性を生かしたOSSアプリを
提案・開発するプロジェクト



- SIPProp Ver.2.0
 - B2BUAフレームワーク
- 雷電
 - 後述
- P2P SIP実装実験プロジェクト
 - SIPをP2P的に使う場合、
 - IETFのP2PSIPとは、別物。
- HOTARUプロジェクト
 - IMS/SIPv6参照実装開発プロジェクト
 - with WIDEプロジェクト
- 俺流プロトコル実装入門
 - プロトコル実装解説書

何がやりたいのか？

● SIPとは？

- 上位にある様々なネットワーク・アプリケーションのためにセッションの管理を行うための「汎用セッション・プロトコル」

● 現状では？

- IP 電話に付随する通信プロトコルとしての「シグナリング・プロトコル」

⇒ 汎用セッション・プロトコルとしてのSIPの復権

なぜ、セッション・プロトコルを？

- 無線ネットワークのオープン化
 - 携帯できる無線デバイス(≠携帯電話)の登場
 - WiMAXやらフェムトセルやらの盛り上がり
 - AndroidやiPhone SDKの登場
 - 端末の解放が見えてきた

- アドホックなネットワークの出現
 - 確実にネットワーク接続されているとは限らないし、局所的なデータ処理だけでできればよい
 - 車車間通信による、事故の回避用のデータなど

⇒ ノード間(クライアント間)の通信となるため、セキュリティが重要！！！！

想定する層関係

アプリケーション層	アプリケーションのための機能を提供する NGN+IMSの「アプリケーション基盤」に相当
セッション層	セッション管理や認証、パーミッションを提供する NGN+IMSの「サービス・ストラタム」に相当
トランスポート層	Peerやリソースへのリーチャビリティを提供する NGN+IMSの「トランスポート・ストラタム」に相当
ネットワーク層	ネットワークを提供する NGN+IMSの「各アクセス網」に相当

いいたいことを！

● SIPProp勉強会

- プロトコル屋やネットワーク屋が、しゃべる機会がない！！！！

- 愚痴を言う機会ともいう(;^_^A アセアセ・・・

- 月に一回のペースで開催中

- 次回は明日！一言言いたい人は、是非！

- ATL Systems社さん(in新宿)が、会場を無償提供

● P2P SIP勉強会

- ネットワークについて、一言物申す勉強会

- 詳細は、こちらへアクセス！

- <http://www.p2psip.jp/>

雷電紹介編

● クライアント間マッシュアップ用 B2BUAアプリケーションサーバ

● VoIP(SIP)とWeb(HTTP)をつなぐもの

- VoIPとWebの相互接続を解決する

● UA、B2BUAベースフレームワークの実装

- クライアント間マッシュアップが可能

● SIPPropをベースとして実装

- オープンソース(Apache License2.0)
- Java実装

自己紹介

- 氏名：鈴木雄介(すずきゆうすけ)
- 年齢：32歳＋3ヶ月
- IT業界歴：約10年
- SIP歴：約0.5年
- 得意言語：Java
- 得意分野：Web、エンタープライズ

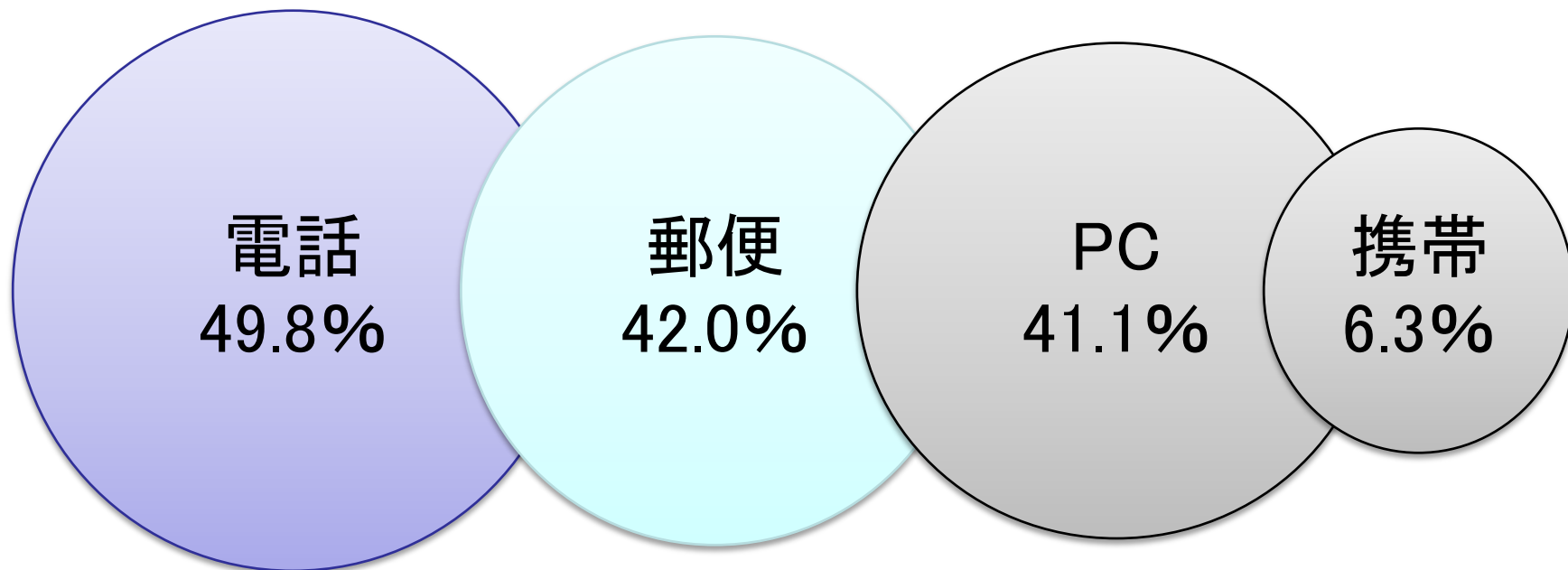
- 特記事項：

- VoIPカンファレンスですが、普通にWebアプリケーションエンジニアとして話します
- なので、VoIP業界では言ってはいけなようなことについてもKYなので、よろしくです
- そもそも「クライアント間マッシュアップ用B2BUAアプリケーションサーバ」ってなに？
- いまだに雷電の実体はよく分からない
- 今日は、僕なりの解釈を話します。チーム内でも認識は違う
 - でも、どこかでつながっているはず

コミュニケーションの未来

- 電話はなくなる。もちろんWebも
 - どうやって融合していくかがテーマ

<2005年通信販売における受注メディア比率(複数回答可)>



- Webアプリエンジニアにとって音声は聖域
- 電話とWebのすれ違い
 - プロトコルのすれ違い
 - ステートレスなHTTPとステートフルなSIP
 - アプリケーションのすれ違い
 - Click2Callは電話クライアントをブラウザに埋め込んでいるだけ。アプリが融合しているわけではない
 - CTI(Computer Telephony Integration)は、プロダクトとして完成されすぎ。
- エンジニアのすれ違い
 - 「それ、ベストエフォート？」
 - 「SIPって音質どーなんすか？」

- どうすれば電話とWebは融合できるのか
 - マルチコンタクトセンターはCTIの延長に過ぎない
- もっとオープンに、もっと自由に可能性を追求するためのプラットフォームが必要
- プラットフォームに求められること
 - リーチが長いこと
 - 標準的な手法であること
 - コネクティビティが高いこと

- 雷電を実現する3つの技術
 - COMET
 - プロトコルハンドリング
 - EDA

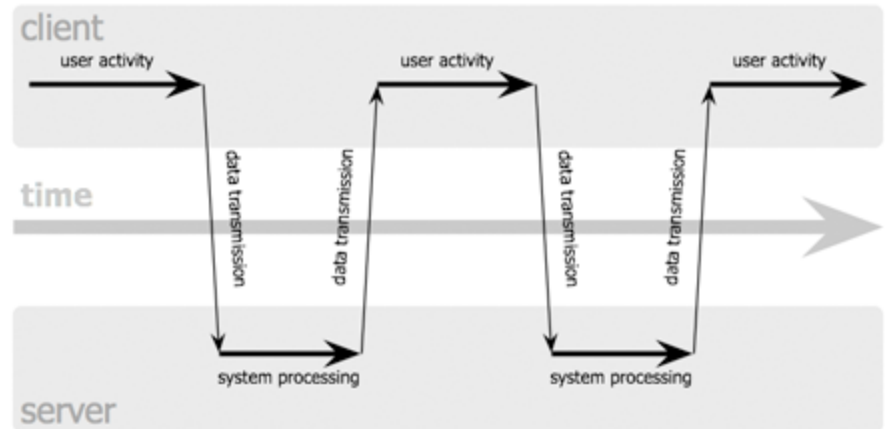


COMET

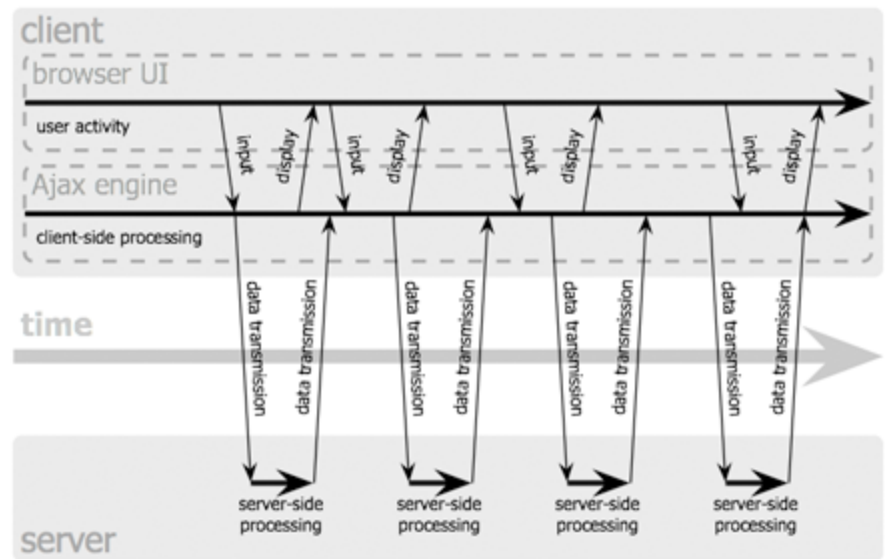
● Ajax

- 通信と画面の描画を非同期に実行することで、通信時間を意識させないようなアプリケーションを構築できる

classic web application model (synchronous)



Ajax web application model (asynchronous)



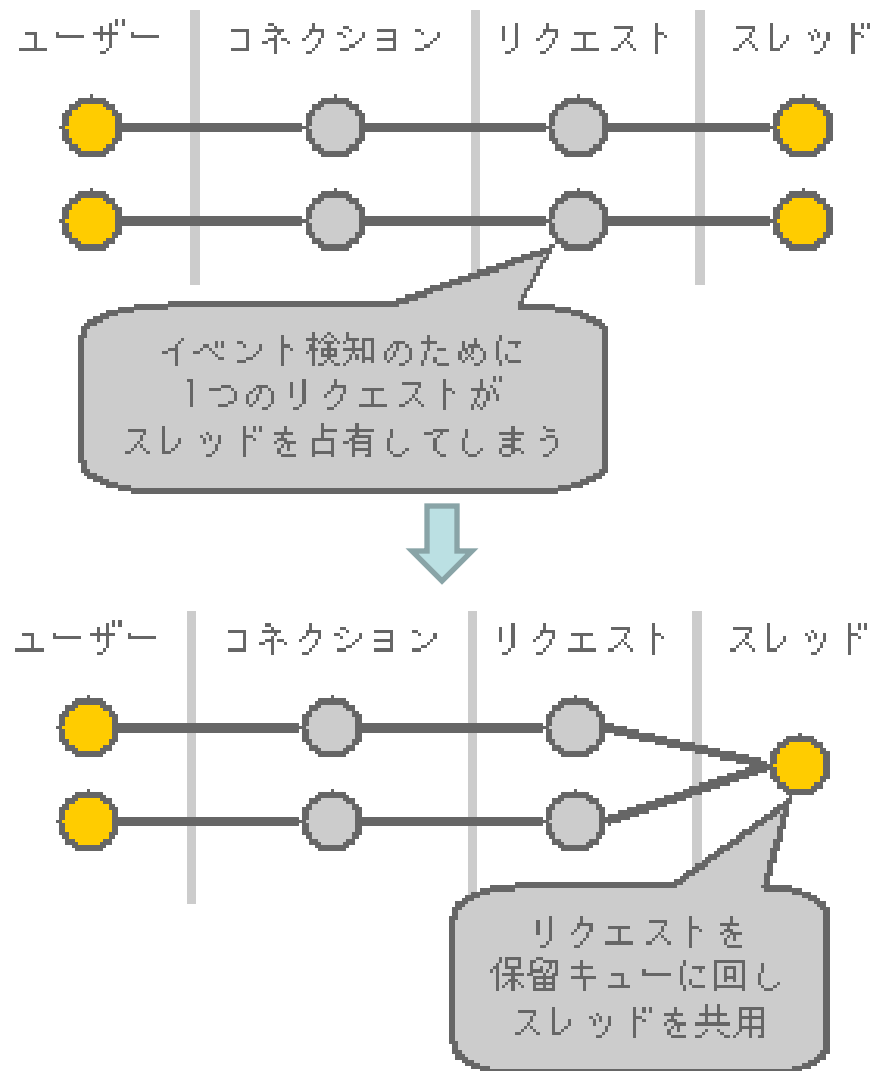
Jesse James Garrett / adaptivepath.com

<http://adaptivepath.com/ideas/essays/archives/000385.php>

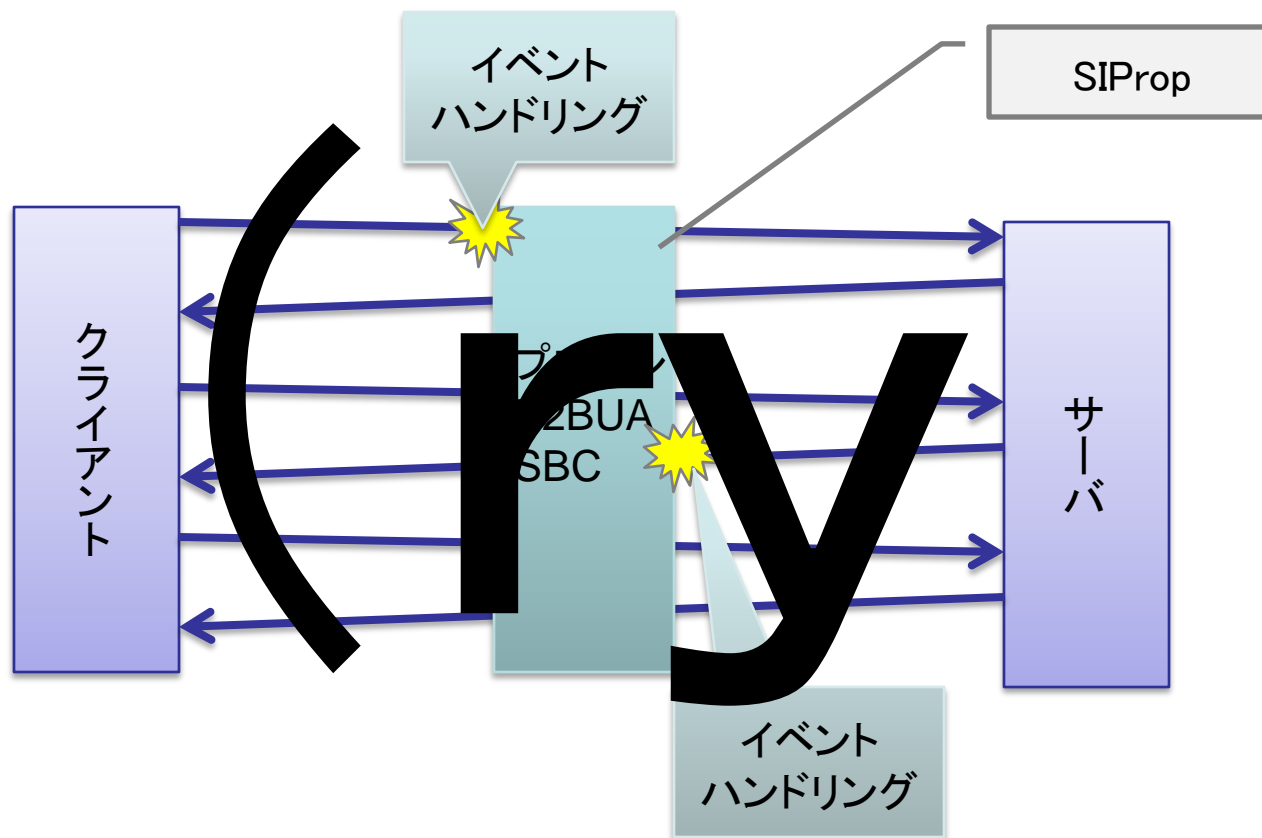
● 非同期IO

- サーバ側でリクエストにスレッドを割り当ててしまうとリソースの無駄が発生する
- そこでイベントが発生しない限りはリクエストを保留キューに置いてスレッドは共用する

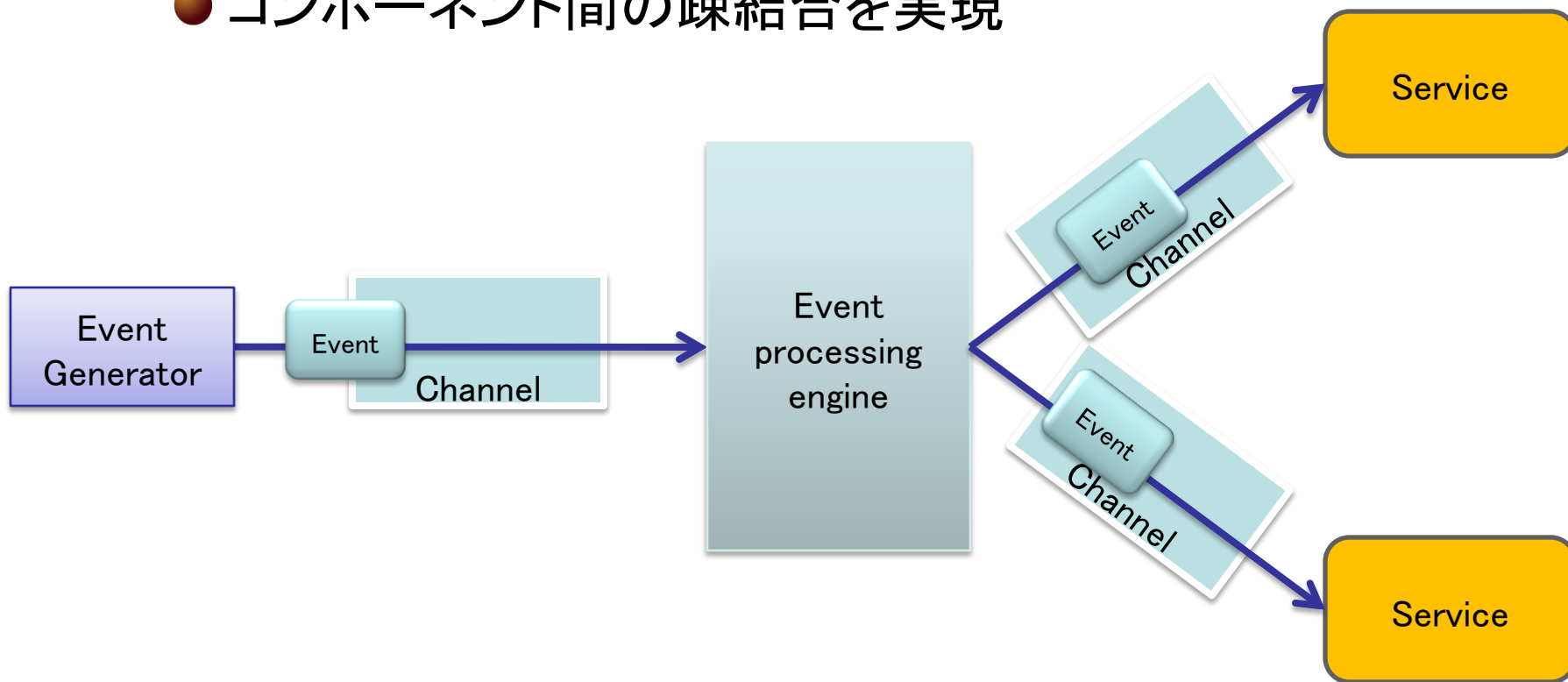
● 今回はApache Tomcat 6で実現



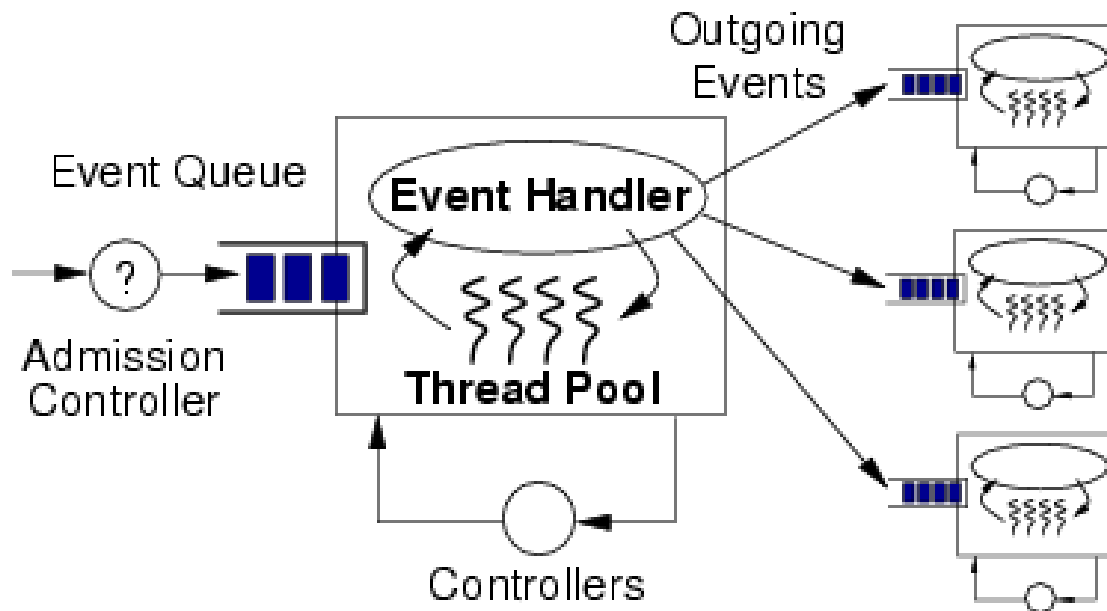
● プロトコルのハンドリング



- EDA (Event-Driven Architecture)
 - イベントを通じてSenderとReceiverを分離
 - IdentityとLocationも分離
 - コンポーネント間の疎結合を実現

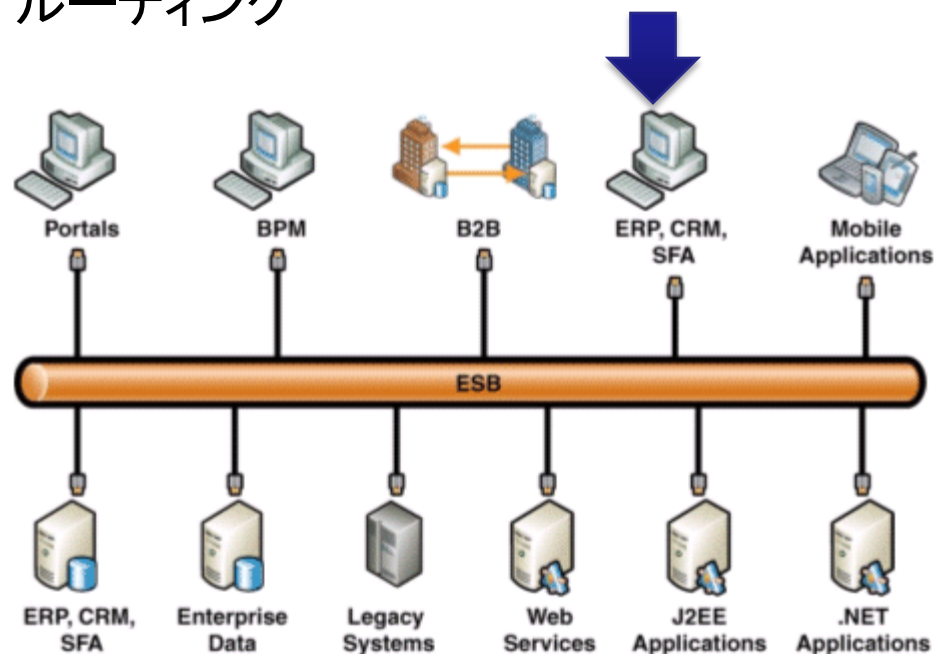
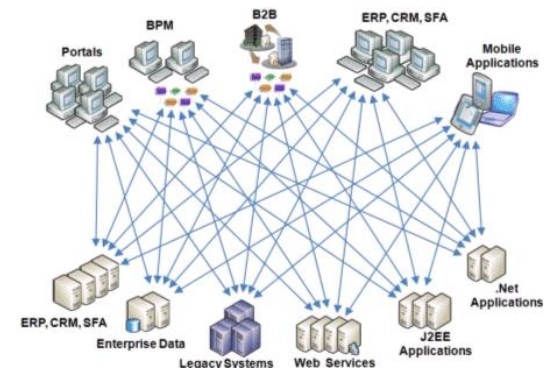


- SEDA (Staged Event-Driven Architecture)
 - ステージに区切ってスレッドプールを実装する
 - ステージ間はイベントドリブンで会話する
- 全体としてのスループットが調整可能になる



● ESB (Enterprise Service Bus)

- 中央にメッセージングバス
- メッセージの様々な交換
 - HTTP、SOAP、JMS
 - パブリッシュ&サブスクライブ、ストア
フォワードメッセージ、ルーティング
 - 同期、非同期

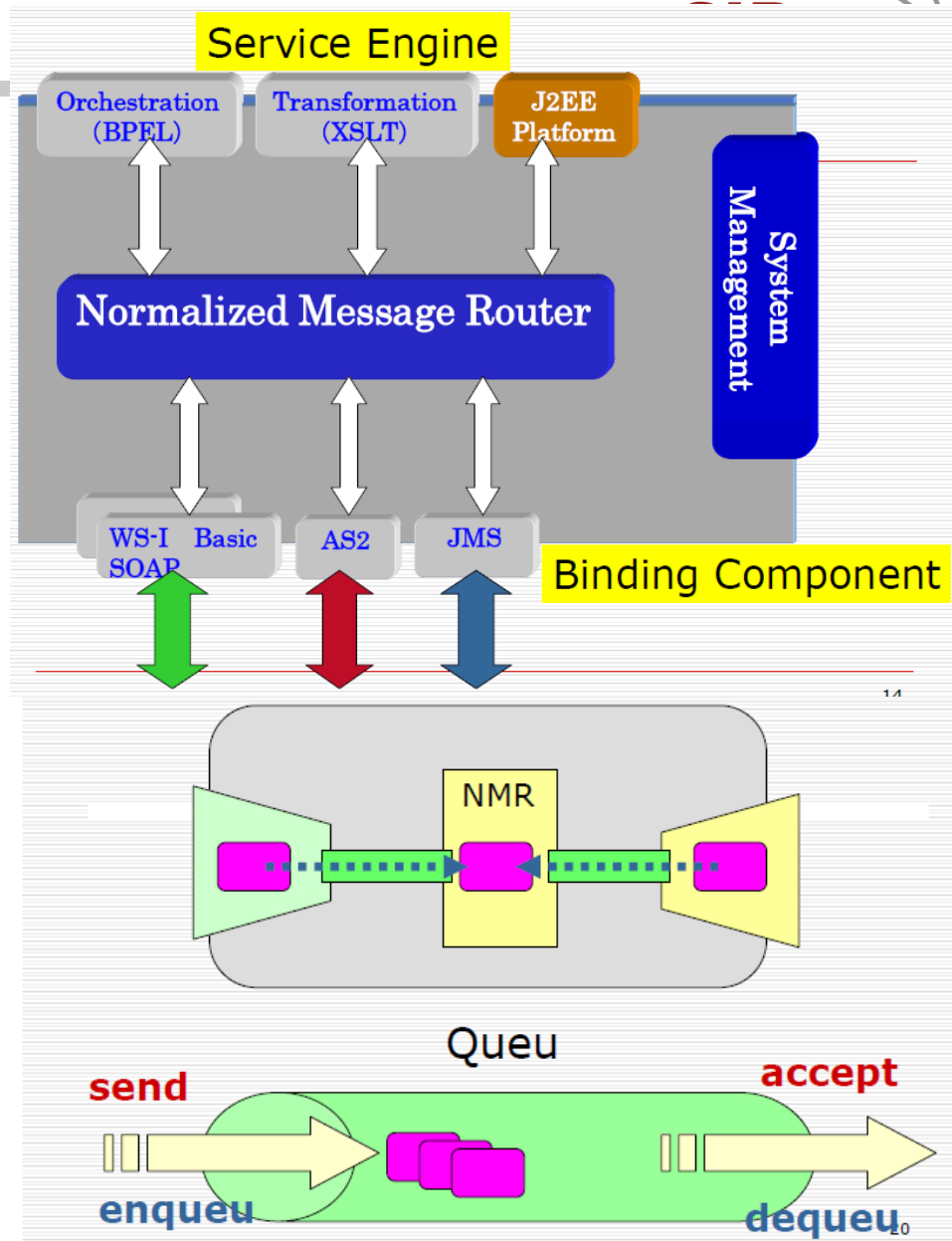




EDA

● JBI

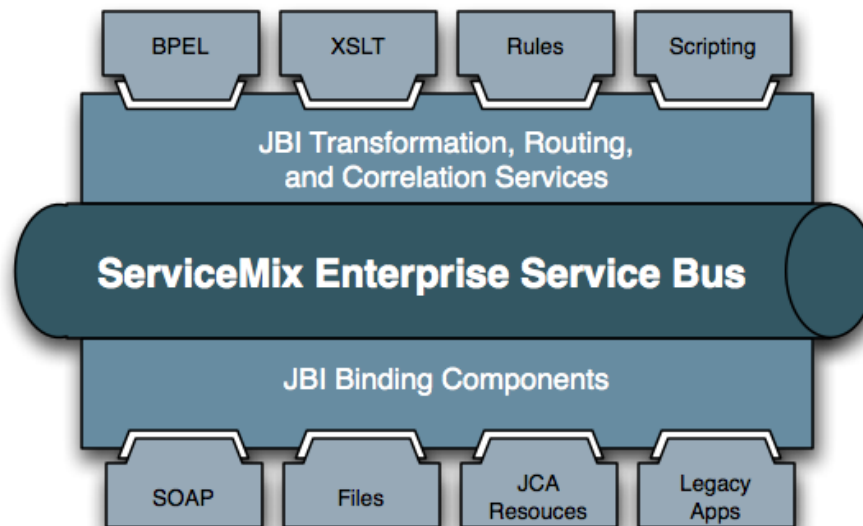
- JavaにおけるESBの標準仕様
- JBIのコンテナがバスとして機能する
- バスの中ではNMR (Normalized Message Router) が中央にいて、登録されたコンポーネント間のメッセージのルーティングする



14

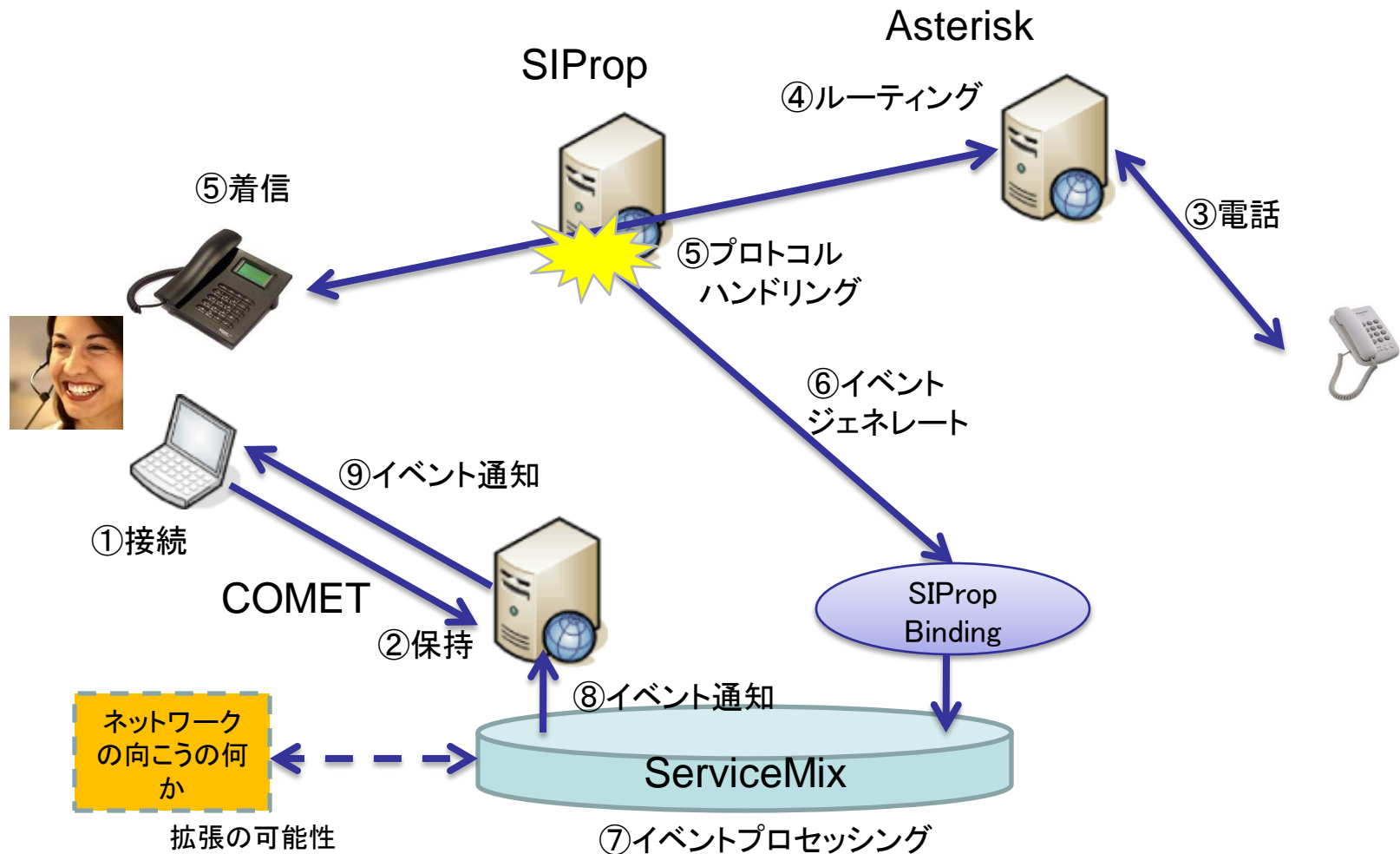
● ServiceMix

- <http://servicemix.apache.org/home.html>
- JBI (Java Business Integration) の実装
- SEDAを実装したESBエンジンを搭載
 - JBI上に登録された各コンポーネントをステージとして扱う
 - コンポーネント間のやり取りはJBIの仕様上でイベントドリブンに処理が行われる



- プラットフォームに求められること
 - リーチが長いこと
 - COMETは、ほとんどのブラウザが対応
 - 標準的な手法であること
 - プロトコルハンドリングは、公開された情報で、かつアプリケーションに非依存でアクセス可能
 - コネクティビティが高いこと
 - EDAによってモジュールを疎結合に保ったまま連携可能
 - SOAとしての連携が可能。分散環境へのサポート

● 僕の中での雷電: COMET + SIPProp + ServiceMix



デモ見てね(はあと)

- マッシュアップによる新しい価値の創出
 - それが何かは、今は分からない
- たとえば、
 - マルチプロトコル
 - XMPP (Jabber)、SMTP、FTP、SOAP、XML、JMS
 - マルチアプリケーション
 - PBX、ケータイ、動画、GIS、エンタープライズ、Web2.0
 - マルチタレント
 - 電話、Web、デザイナー、スーツ

雷電の裏事情

● 某ネットワークの不透明性

- 承認が必要？
- 利用料は？

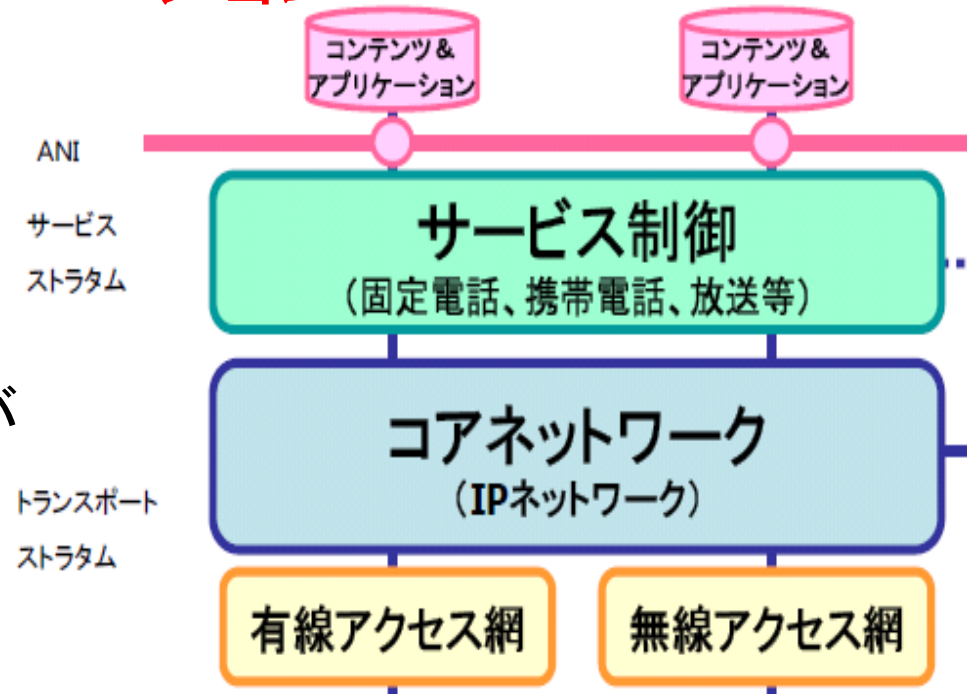
⇒ **ベストエフォートバージョン**

● 雷電

- コンテンツ
アプリケーションサーバ

● SIPProp

- サービス制御



出典: 沖電気 千村様公開プレゼン資料より

ご静聴ありがとうございました。

<(_ _)>

<http://www.siprop.org/>

Blog: <http://noritsuna.siprop.org/>