ROA

Resource Oriented Architecture
リソース指向アーキテクチャー

日本アイ・ビー・エム株式会社
クラウド・エバンジェリスト
米持 幸寿
pandrbox@twitter

米持 幸寿

日本アイ・ビー・エム株式会社
クラウド・エバンジェリスト

- 自然言語解析
- Big data
- クラウド・コンピューティング

これまで取り扱ってきた
テクノロジー

Web2.0、リッチクライアント、
SOA、J2EE/EJB、
Webサービス、XML、Java

リソース指向とはなにか

- 正統派の「RESTful」の思想
  - Webのようなサービス
    - スコープ情報はURI に
    - メソッド情報はHTTPメソッドに
- Roy Thomas Fielding 氏の論文に端を発する
  - http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
- オライリー「RESTful Webサービス」で語られている
  - 標準ではない
  - ルールではない

簡単にいうと

- URI（URL）で「リソース」を示す（identity）
  - 「なにを」
  - アドレス可能
- HTTP で「メソッド」を示す
  - 「どうしたい」
  - ステートレス

アドレス可能性

- リソースをURIで表現できる
  - http://www.cloud.com/INSTANCE/123456
  - ftp://ftp.theses.com/doc?search=ROA
- アドレス可能でないもの
  - 多くのAjaxアプリケーション
  - 多くのFLASHアプリケーション
  - キャッシュできない

制約された統一インターフェース

HEAD	特定リソースのメタ情報を取得
GET	特定リソース状態（コンテンツ）を取得
PUT	特定リソース状態の書き込み
DELETE	特定リソース状態を削除（除去）
POST	リソースを追加
OPTIONS	利用できるメソッド一覧の取得

CRUD（Create Read Update Delete）を実現する

ステートレス

- リクエストを連続して呼び出す必要がない
  - 必要なパラメーターはすべて送信する
- 状態をサーバーが維持しない
  - 状態の管理はクライアント側の責任

SOAPのときも散々議論された

安全性 と べき等性

- 安全性
  - OPTIONS、HEAD、GET は、リソース状態に影響を与えない
    → 平たく言うと「リードオンリー」
- べき等性
  - OPTIONS、HEAD、GET、PUT、DELETE は、1回以上呼び出しても同じ結果となる
- POSTは別格である

注！ ↑・・のように作らなくてはだめよ、ということ

中身はXMLか？

- 中身がなにであるか、は議論の中心ではない
  - HTML
  - 画像・動画・音声
  - XML
  - JSON
  - テキスト

例

ROAな例：APP

Atom Publishing Protocol

- The Atom Syndication Format：RSSに代わるフィードフォーマットとしてIETFで標準化
- APPは、ATOMを利用して記事を投稿（Publish）するプロトコルとしてIETFで標準化

明言はしてないが、思想はAtom on ROA

http://tools.ietf.org/html/rfc4287
http://www.ietf.org/rfc/rfc5023.txt
http://abdera.apache.org/

WADL

Web Application Description Language

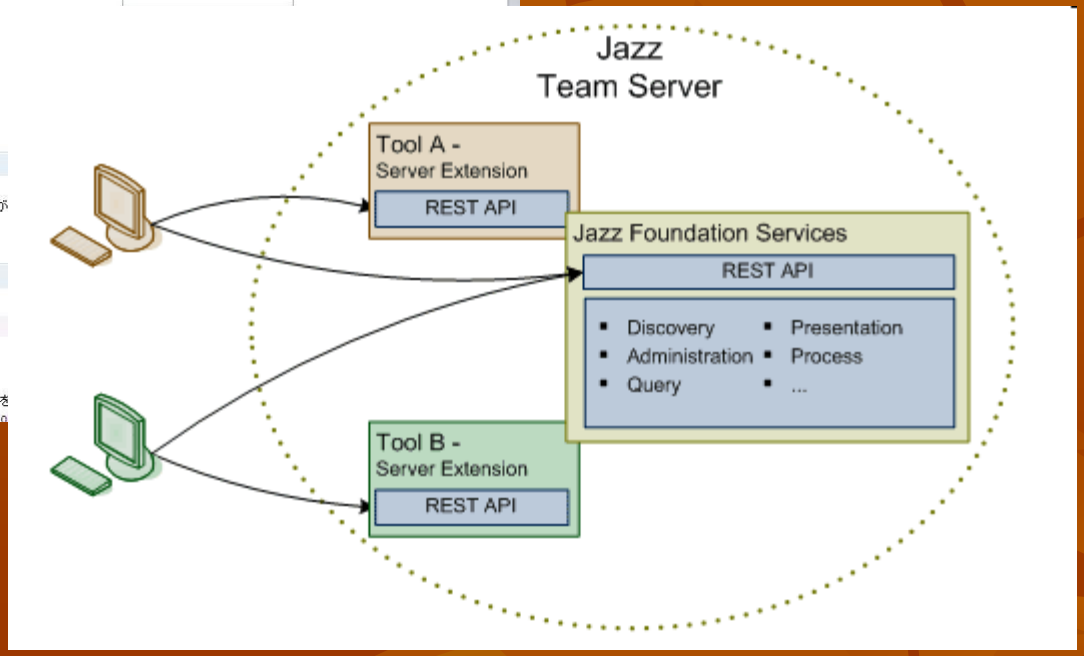
- RESTfulサービスの記述言語
  - SOAPにとってのWSDLみたいなもの
- クライアント実装
  - Java™ http://wadl.dev.java.net
  - Ruby http://www.crummy.com/software/wadl.rb/

http://www.w3.org/Submission/wadl/

Open Services for Lifecycle Collaboration
Lifecycle integration inspired by the web

ROAな例:OSLC

Jazz / Rational Team Concert

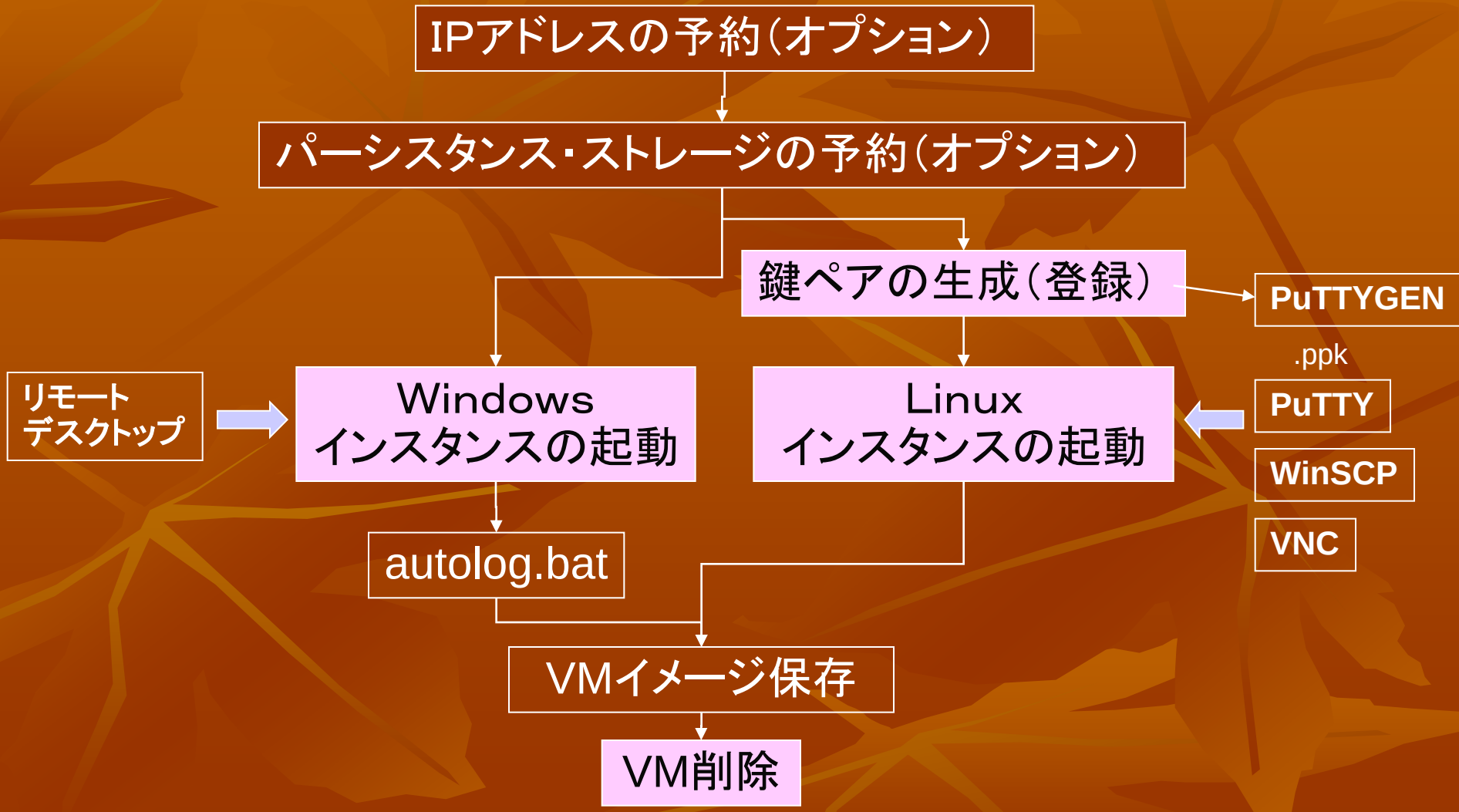


http://open-services.net/

https://jazz.net/wiki/bin/view/Main/ResourceOrientedWorkItemAPIv2

ROA な例：IBM SBCE

Smart Business Cloud：操作の概要

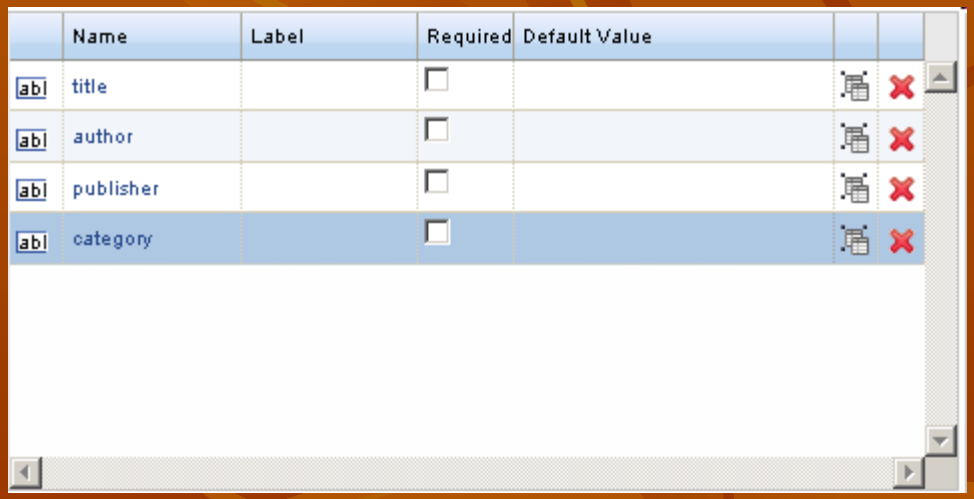


REST API 一覧

- Image Management
  - GET /offerings/image
  - GET /offerings/image/$IMAGE_ID
  - GET /offerings/image/$IMAGE_ID/agreement
  - DELETE /offerings/image/$IMAGE_ID
- Instance Management
  - GET /instances
  - GET /instances/$INSTANCE_ID
  - POST /instances
  - DELETE /instances/$INSTANCE_ID
  - PUT /instances/$INSTANCE_ID
    - PUT expiration Time
    - PUT state
  - GET /requests/$REQUEST_ID
- Storage Management
  - GET /offerings/storage
  - GET /storage
  - GET /storage/$STORAGE_ID
  - POST /storage
    - POST create storage
    - POST clone storage
  - DELETE /storage/$STORAGE_ID
- IP Address Management
  - GET /offerings/address
  - GET /addresses
  - POST /addresses
  - DELETE /addresses/$ADDRESS_ID
  - GET /offering/vlan
- Key Management
  - GET /keys
  - GET /keys/$KEY_NAME
  - POST /keys
    - POST name
    - POST name & publicKey
  - PUT /$KEY_NAME
    - PUT public key
    - PUT default
  - DELETE /keys/$KEY_NAME
- Location Management
  - GET /locations
  - GET /locations/$LOCATION_ID

ROAな例：Zero Resource Manager

- Project Zero に搭載されているRESTfulサービス公開のためのフレームワーク
- appbuilderで簡単に構築可能
  - データモデルの定義
  - ドライバーの記述



CodeZine「はじめて使うProject Zero」
http://codezine.jp/article/detail/3650

ROAな例：JAX-RS（JSR-311）

- Javaアノテーションを利用して、RESTfulサービスを実装するためのフレームワーク
- JavaEE™ 6に搭載（JAX-RS 1.1）
  - Oracle GlassFish™ Open Source Edition 3.x
  - TmaxSoft の JEUS™ 7
  - IBM WebSphere™ Application Server V8.0 など
- @GETといったアノテーションでRESTを実装可能

その他書籍の中で紹介されているもの

- Ruby on Rails
- Restlet
- Django

参考文献

- Resource Oriented Architecture and REST
  - European Commission
    Joint Research Centre
    Institute for Environment and Sustainability
  - http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/Resource_orientated_architecture_and_REST.pdf

- RESTとROA
  - 株式会社リコー ソフトウェア研究開発本部 山本陽平
  - http://blogs.ricollab.jp/webtech/2008/02/rest_and_roa/

オライリー社 RESTful 2冊

ROAに従わなければいけないか

ROAは常に優れているか

ありがとうございました

- 本資料の多くは、参考文献をもとにまとめたものです。
- 記述されている内容はスピーカーの個人的な意見を含んでおり、必ずしも所属する企業・団体を代表するものではないかもしれません。
- 内容は絶対的かつ確定的なものではありません。状況による例外もありますし、時間によっても変化するでしょう。間違い等への指摘はSNS等で受け付けますが、攻撃は好みません。