

# Push-Enabling RESTful Business Processes

**Cesare Pautasso**

Faculty of Informatics,  
University of Lugano, Switzerland

[c.pautasso@ieee.org](mailto:c.pautasso@ieee.org)

<http://www.pautasso.info>

[@pautasso](#)

**Erik Wilde**

EMC

[dret@dret.net](mailto:dret@dret.net)

<http://dret.net>

@dret

6.12.2011

# RESTful Business Processes

**Cesare Pautasso**

Faculty of Informatics,  
University of Lugano, Switzerland

[c.pautasso@ieee.org](mailto:c.pautasso@ieee.org)

<http://www.pautasso.info>

[@pautasso](#)

**Erik Wilde**

EMC

[dret@dret.net](mailto:dret@dret.net)

<http://dret.net>

@dret

6.12.2011



**BPM**



**REST**

**Business  
Process  
Management**

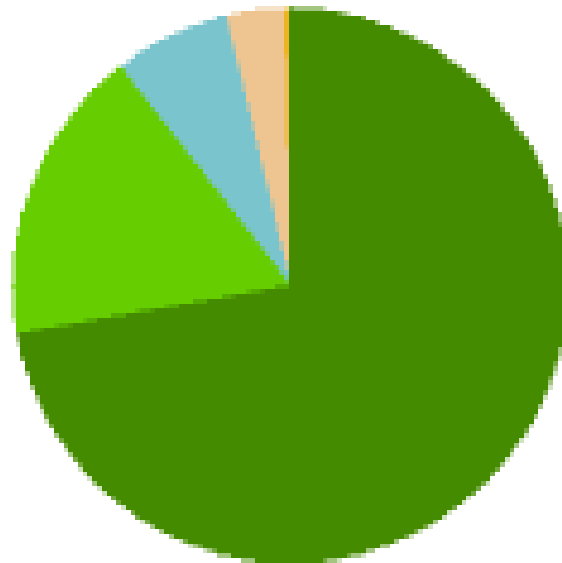
**RESTful  
Web Services**

# RESTful APIs...



# RESTful APIs...

## Protocol Usage by APIs



- REST (72%)
- SOAP (17%)
- JavaScript (6%)
- XML-RPC (3%)
- Atom (0%)

ProgrammableWeb.com 12/05/11



Università  
della  
Svizzera  
italiana



Interoperability

BPM

Management

Metadata

Reliability

Security

Transactions

State

Messaging



## XML Specifications





Interoperability

BPM

Management

Metadata

Reliability

Security

Transactions

State

Can you do it  
with REST?



anno Q

© Deutschland GmbH  
Jasperstraße 11  
40880 Ratingen  
Tel: +49 (0) 21 02 - 27 162-100  
Fax: +49 (0) 21 02 - 27 162-101  
info@annoq.com - www.annoq.com

annoQ Software GmbH  
Gewerbestraße 11  
CH-4300 Cham  
Telefon +41 (0) 41 - 743 00 00  
Telefax +41 (0) 41 - 743 00 08  
info@annoq.com - www.annoq.com

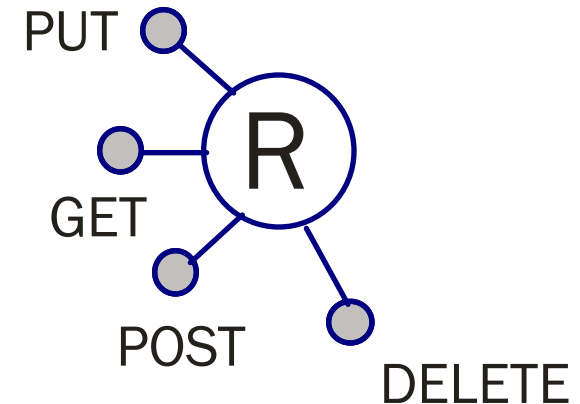
“ We believe there is huge potential to marrying REST with workflow and BPM.

[...]

Combined with the architecture of the Web, a workflow service can provide both a truly **simple, portable, and flexible** way to build workflow driven integrations and applications.

”

- Web Services expose their data and functionality through **resources** identified by **URI**
- **Uniform Interface** constraint: Clients interact with resources through a fix set of verbs.  
Example HTTP:  
GET (read), POST (create), PUT (update), DELETE
- **Multiple representations** for the same resource
- **Hyperlinks** model resource relationships and valid state transitions for dynamic protocol description and discovery

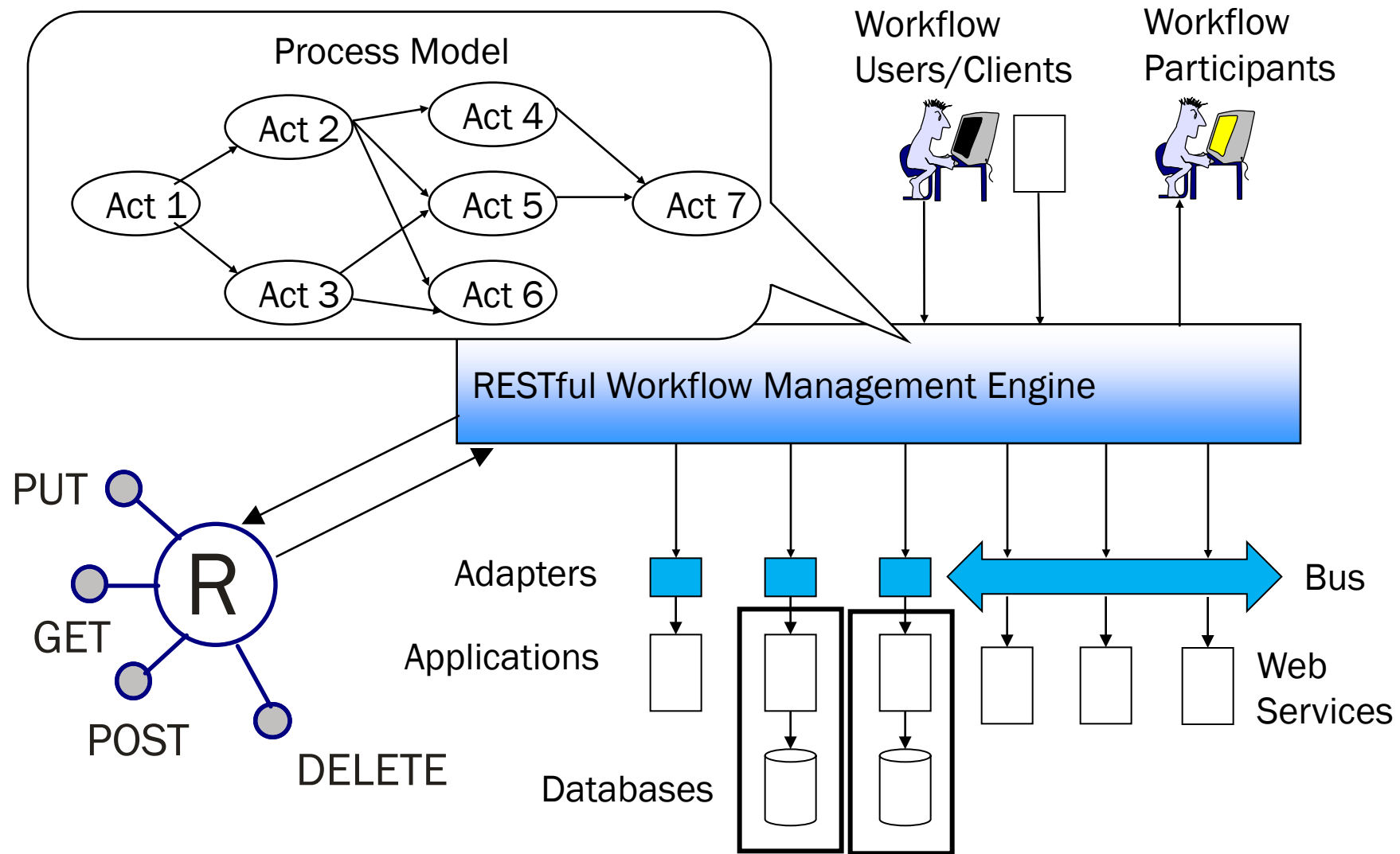


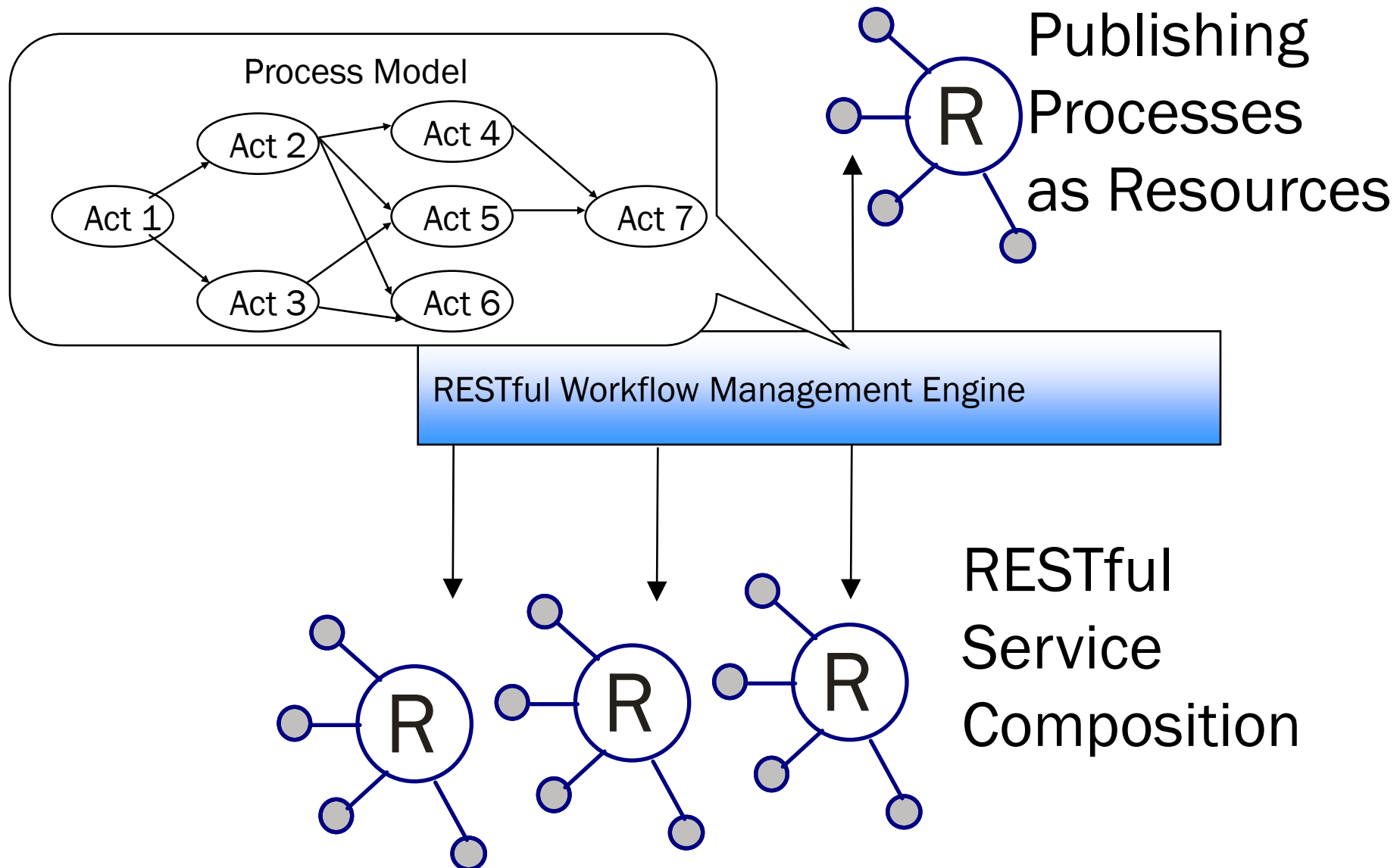
“ We believe there is huge potential to marrying REST with workflow and BPM.

- The HATEOAS (hypermedia and linking) principal of REST is logically a dynamic state machine and **fits very well** with how workflow and BPM systems are designed.
- Combined with the architecture of the Web, a workflow service can provide both a truly simple, portable, and flexible way to build workflow driven integrations and applications.

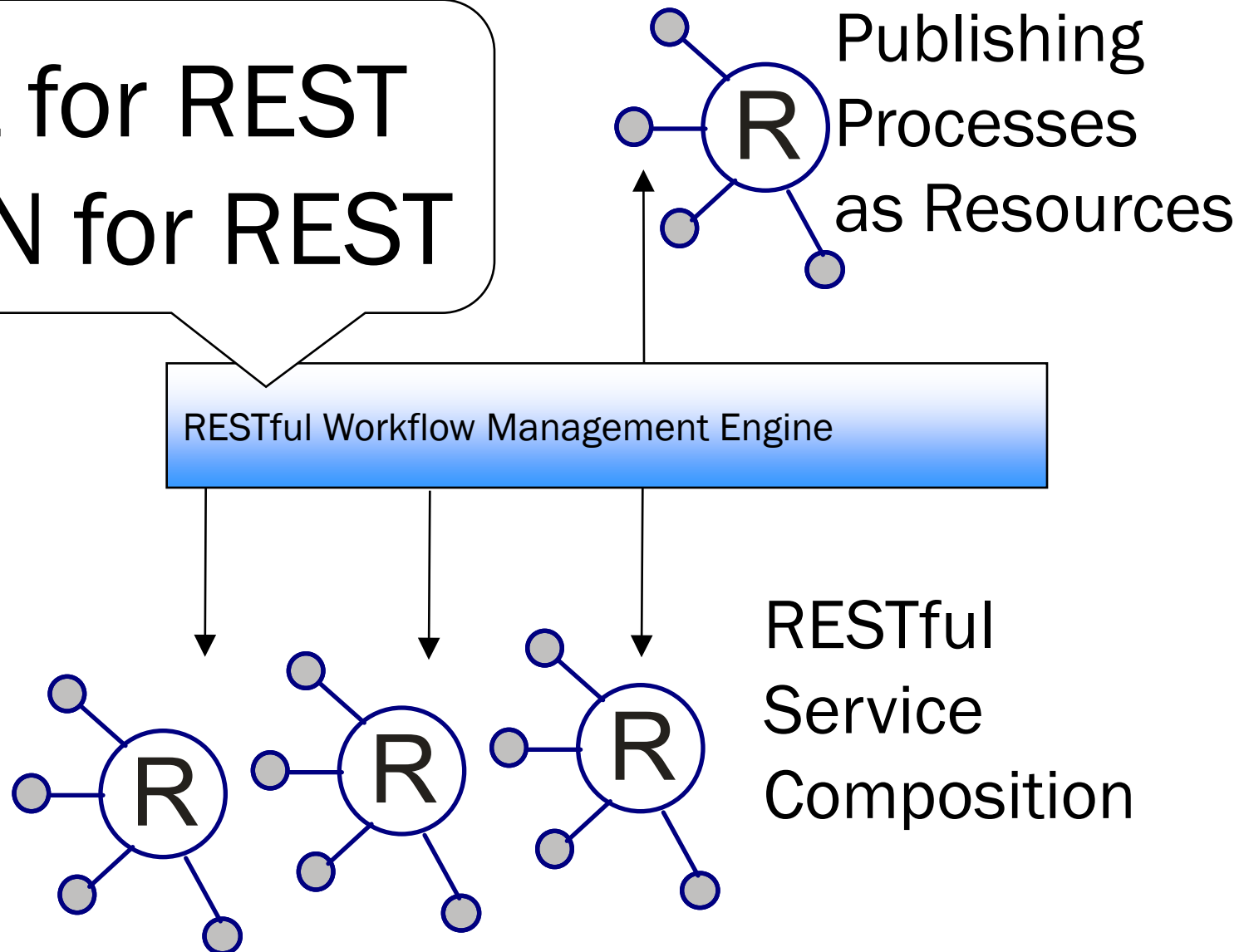
”







BPEL for REST  
BPMN for REST

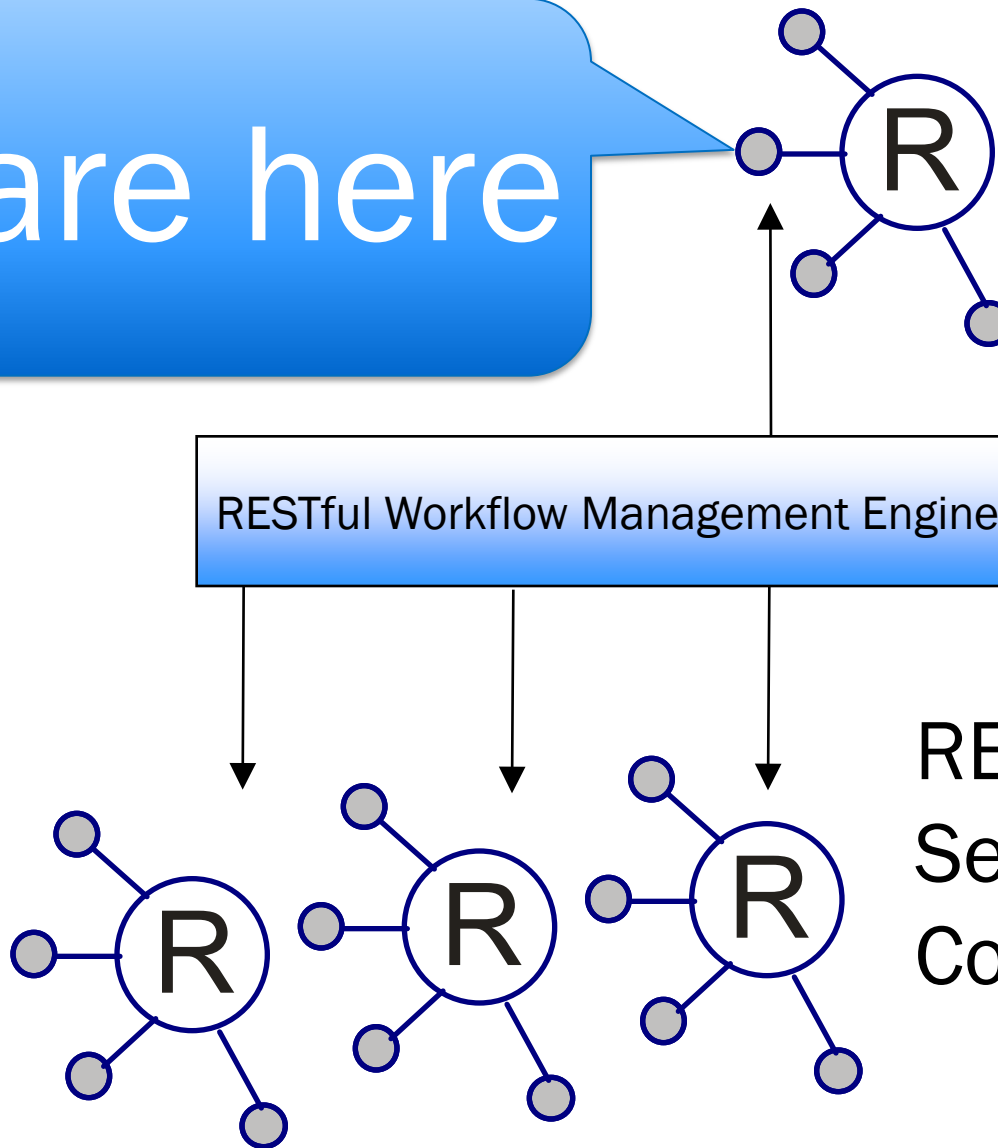


We are here

Publishing  
Processes  
as Resources

RESTful Workflow Management Engine

RESTful  
Service  
Composition





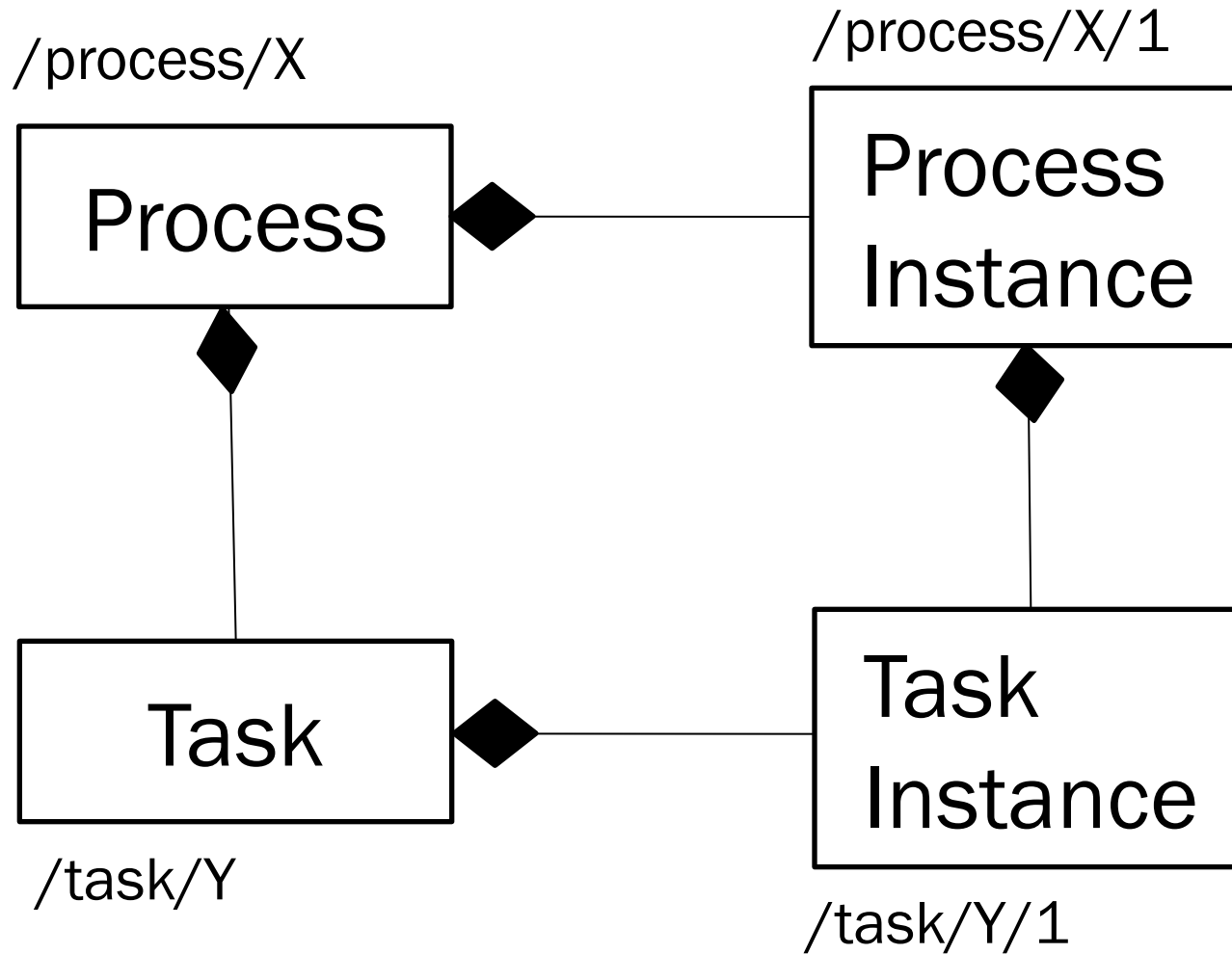
# BPM

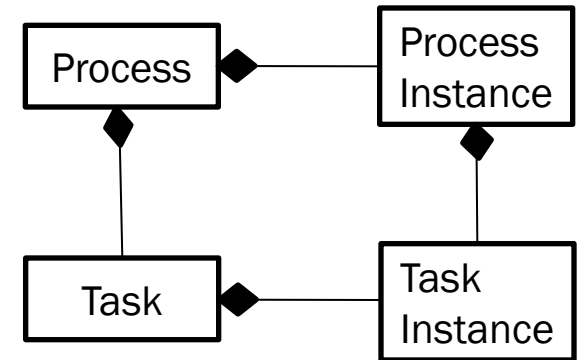
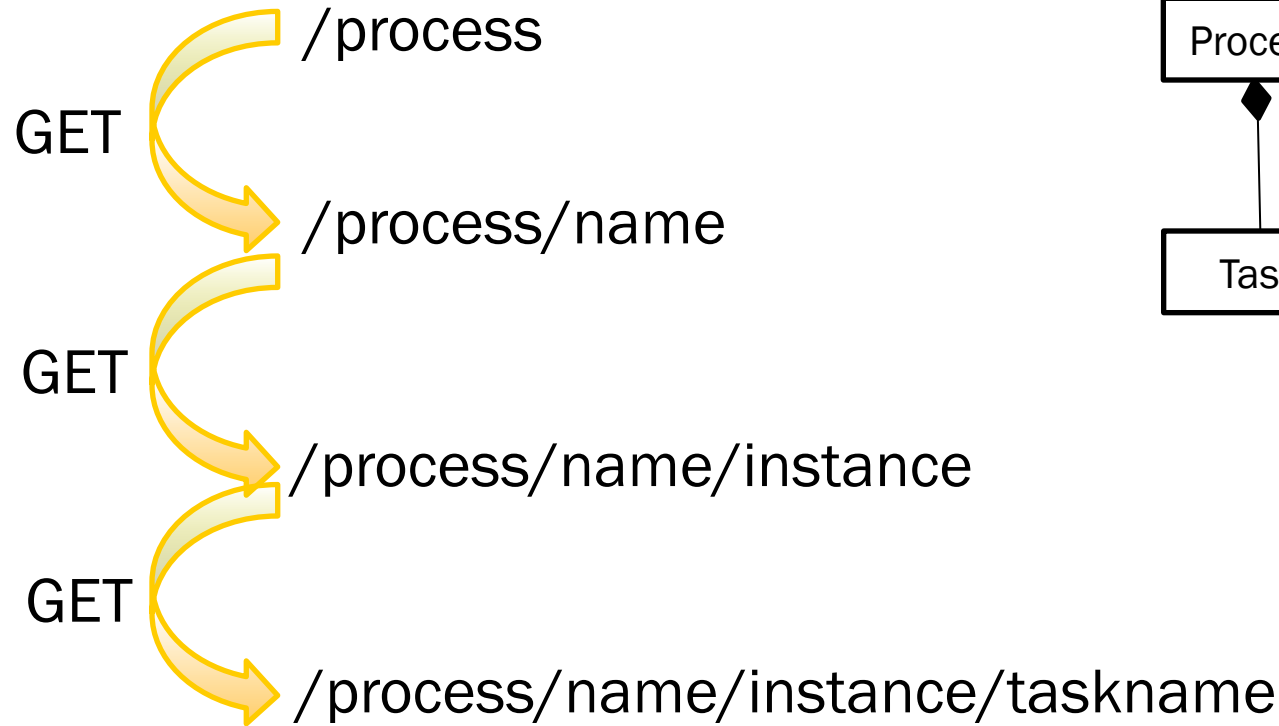
- Processes
- Tasks
- Control Flow
- Data Flow
- ...

# REST

- Resources/URIs
- Uniform Interface
- Representations
- Hypermedia

# Everything is a resource





Follow links to discover the processes  
deployed as resources

# Representations

Web page  
with form to start  
a new process  
instance

ContentType:

text/html

ContentType:

application/bpmn+xml

BPMN2.0 process  
source code

GET /process/name

ContentType:

text/plain

ContentType:

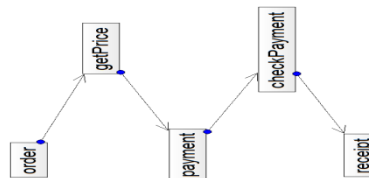
application/json

Basic textual  
description  
of the process

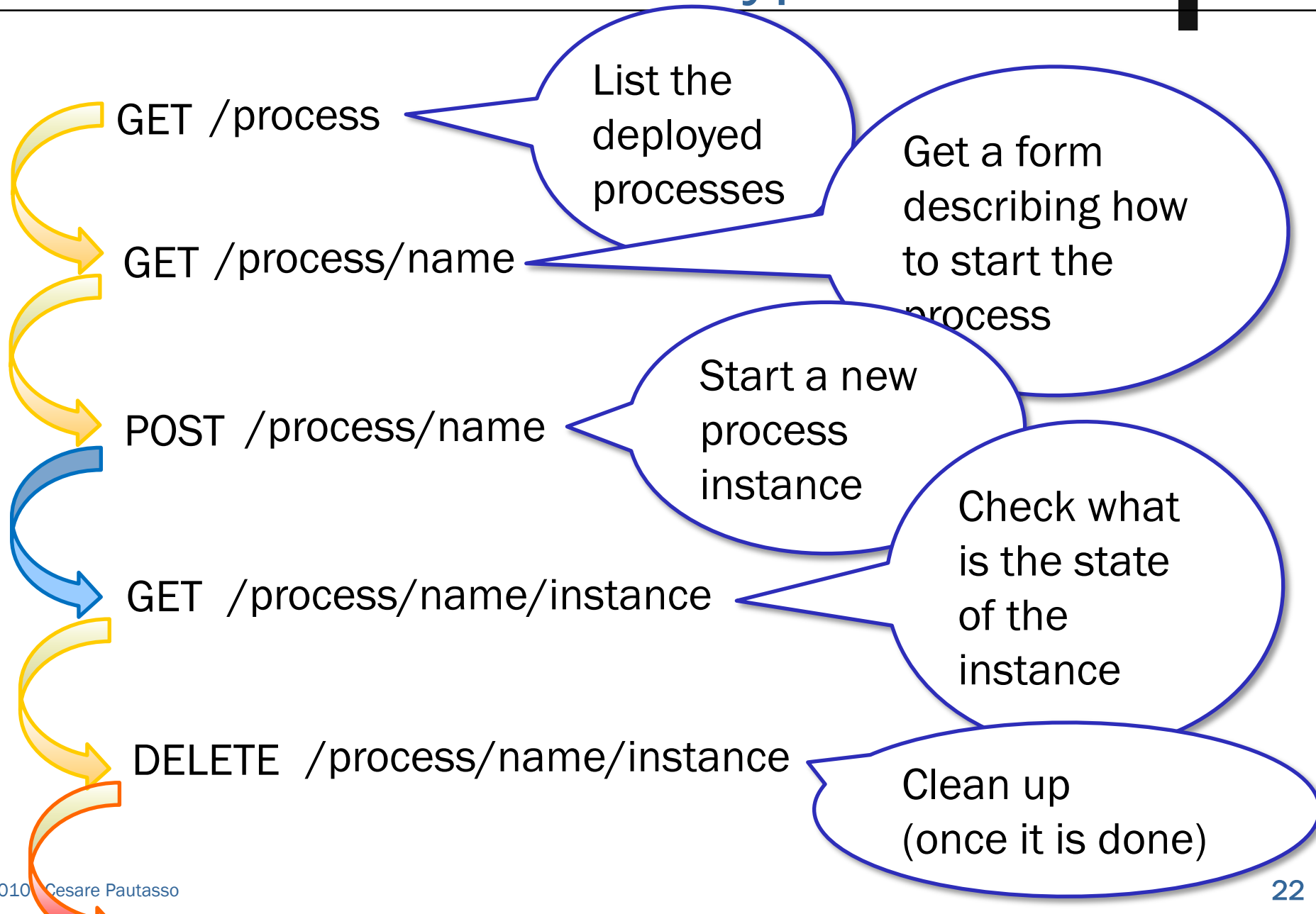
ContentType:

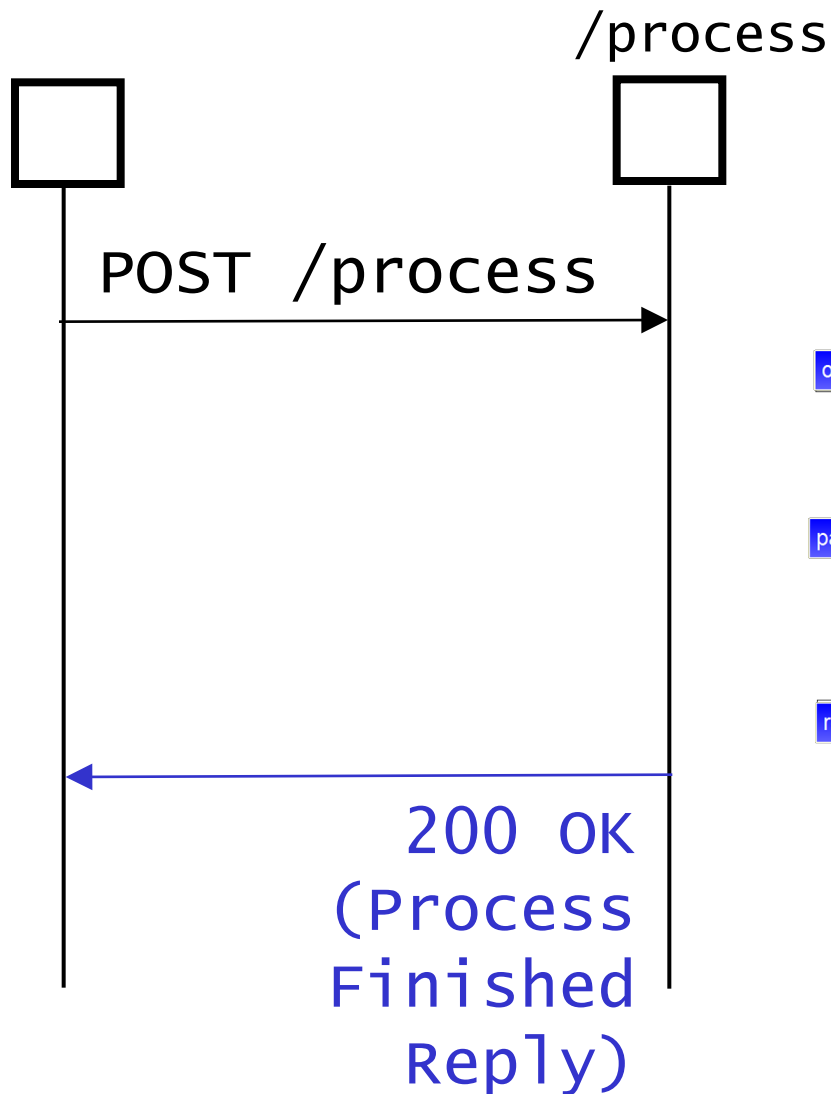
image/svg+xml

Process  
metadata  
in JSON

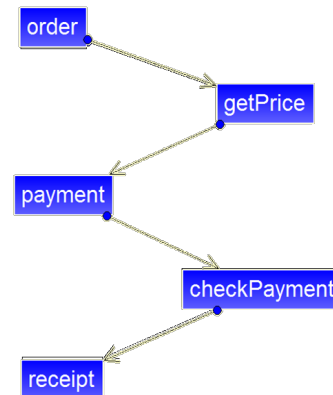




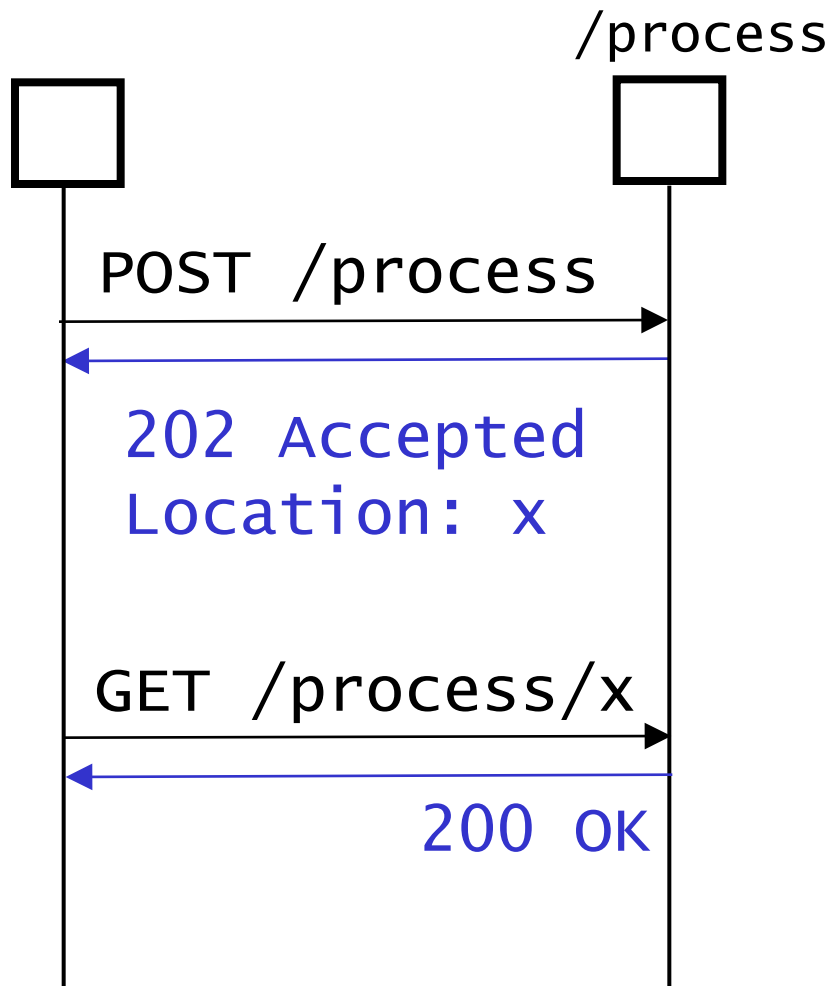




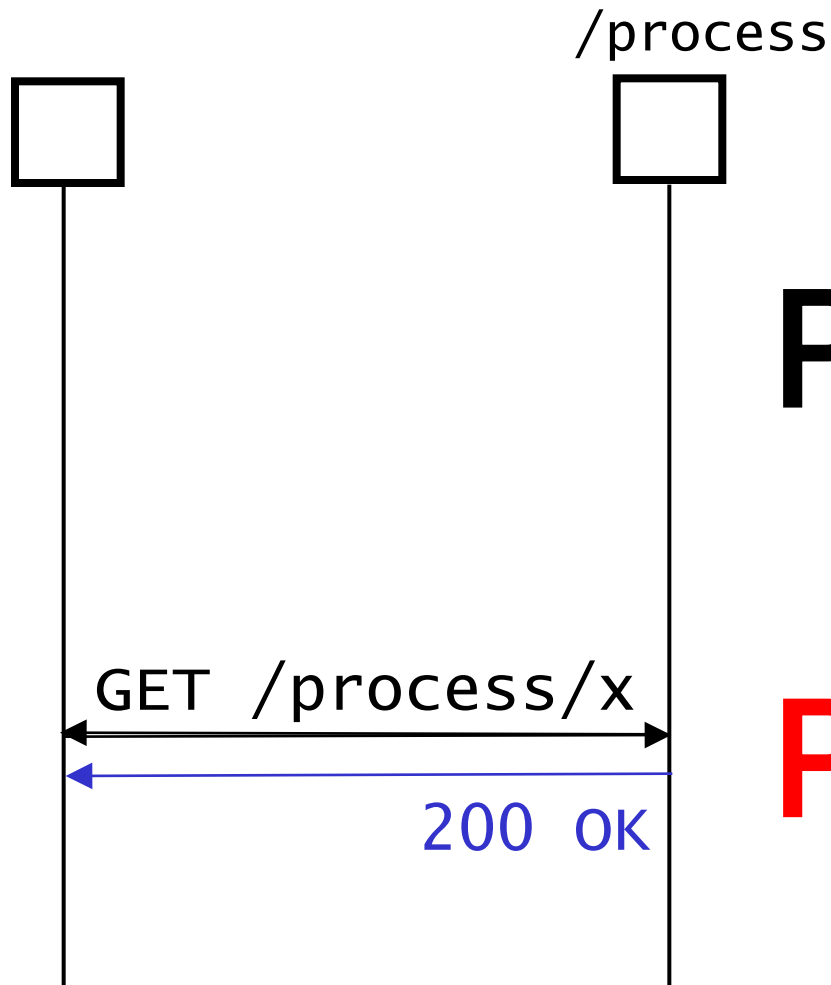
- Should the client be kept waiting for the process to run until completion?



- Clients may want to block until the whole process has completed its execution (or it decides to reply to them)



- The client starting a long running process is redirected to a location *x* representing the newly started process instance
- The process and the client run asynchronously
- The client may retrieve the current state of the process instance at any time



- Problem: how can the process instance tell the client that it has reached a certain state?

## PULL

- Easy to use a PULL-based event notification with HTTP

## PUSH

- Can we also support PUSH-based event notification with HTTP?



# Push-Enabling RESTful Business Processes

**Cesare Pautasso**

Faculty of Informatics,  
University of Lugano, Switzerland

[c.pautasso@ieee.org](mailto:c.pautasso@ieee.org)

<http://www.pautasso.info>

[@pautasso](#)

**Erik Wilde**

EMC

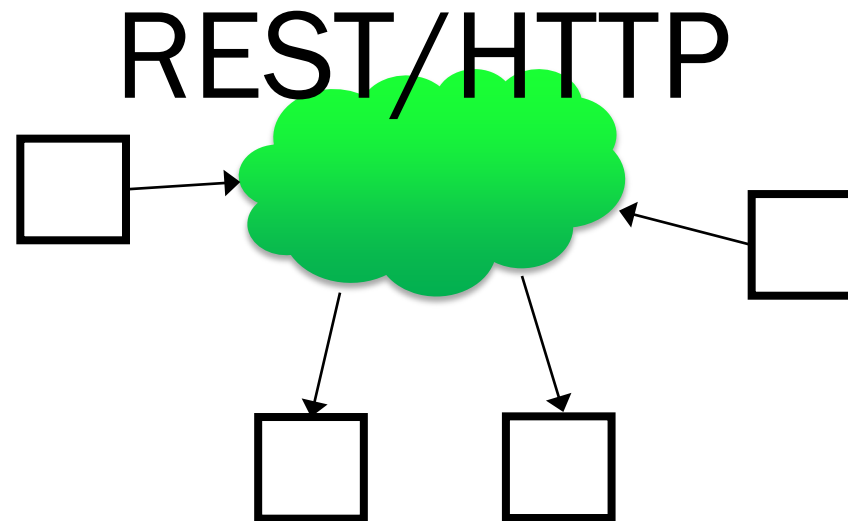
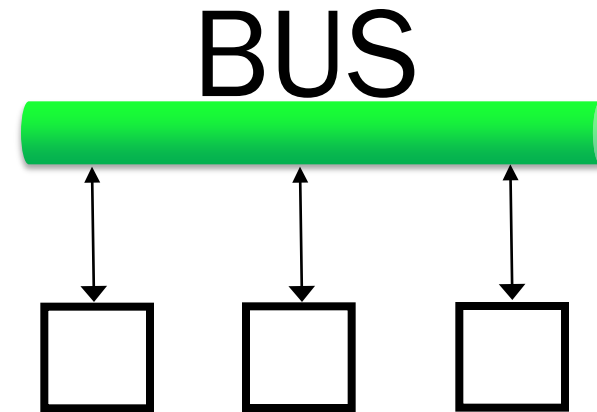
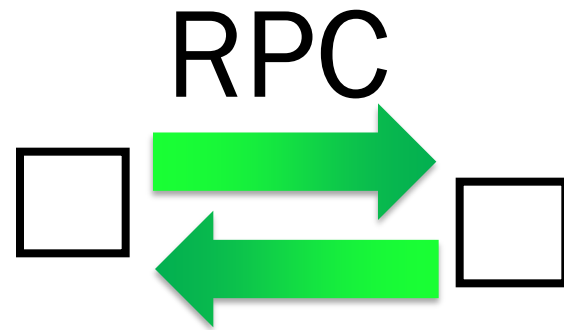
[dret@dret.net](mailto:dret@dret.net)

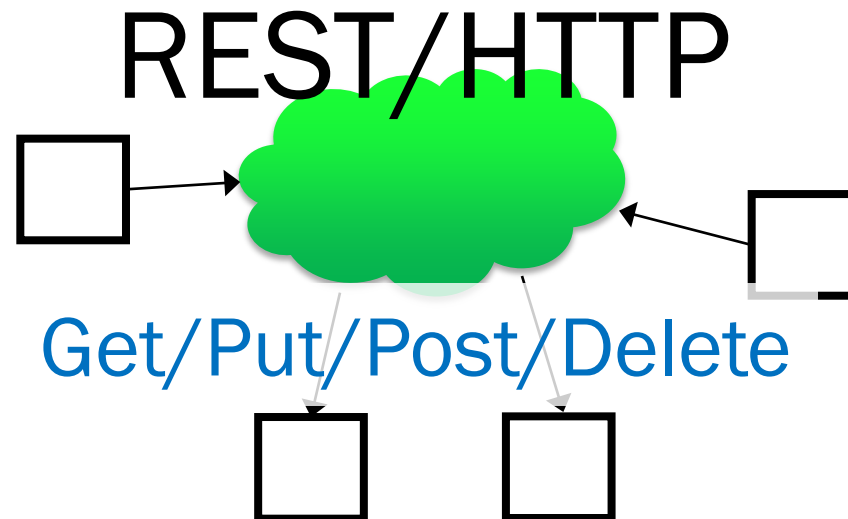
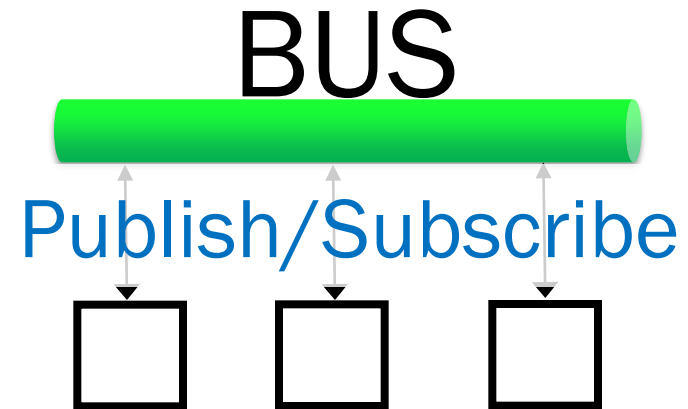
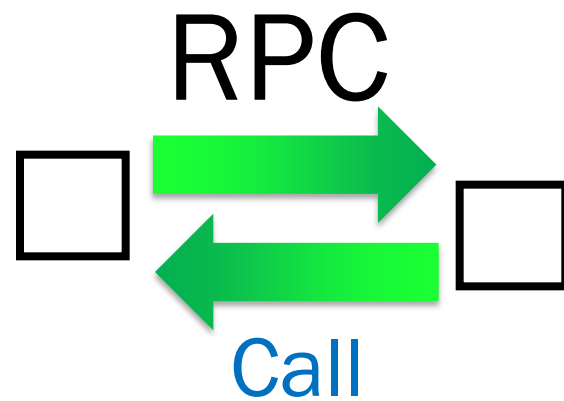
<http://dret.net>

@dret

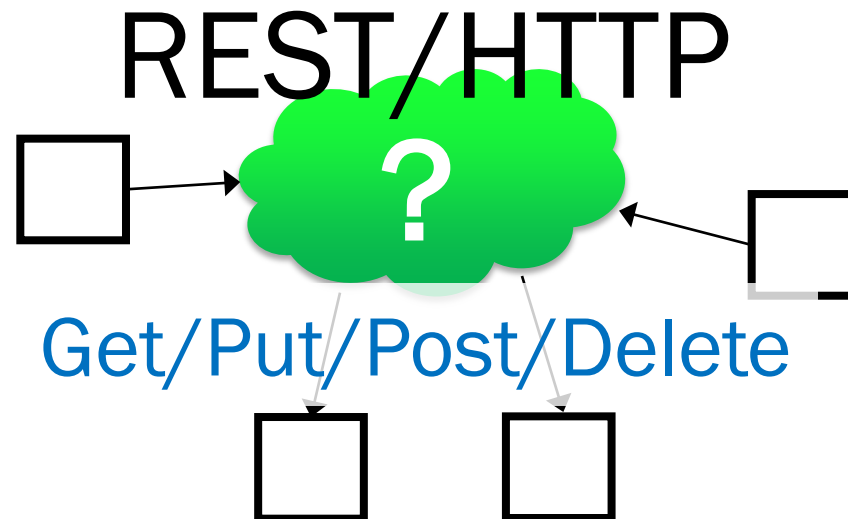
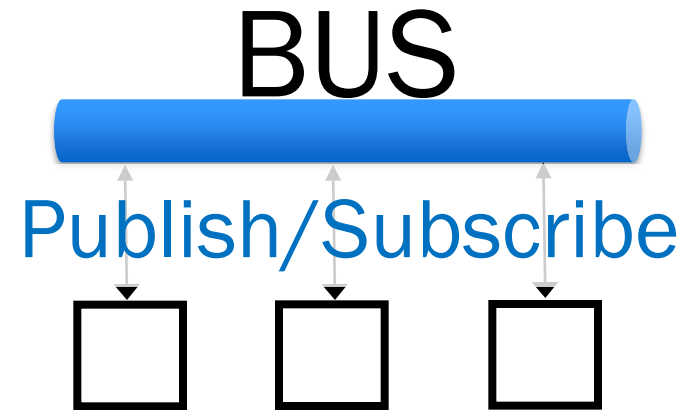
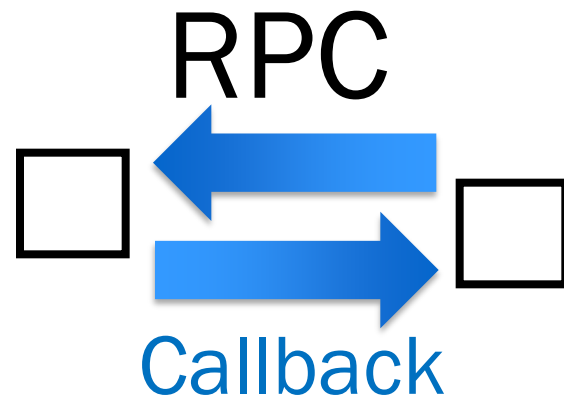
6.12.2011

# What is your SOA connector?





# What about event notifications?



1. Web Feeds (PubSubHubbub)
2. HTTP Long Polling
3. Inverted REST (HTTP Callbacks)
4. WebSockets
5. (XMPP)

GET /process/name

ContentType:

application/atom+xml

Web feed representing  
the collection of  
process instances with  
links to each instance

GET /process/name/instance

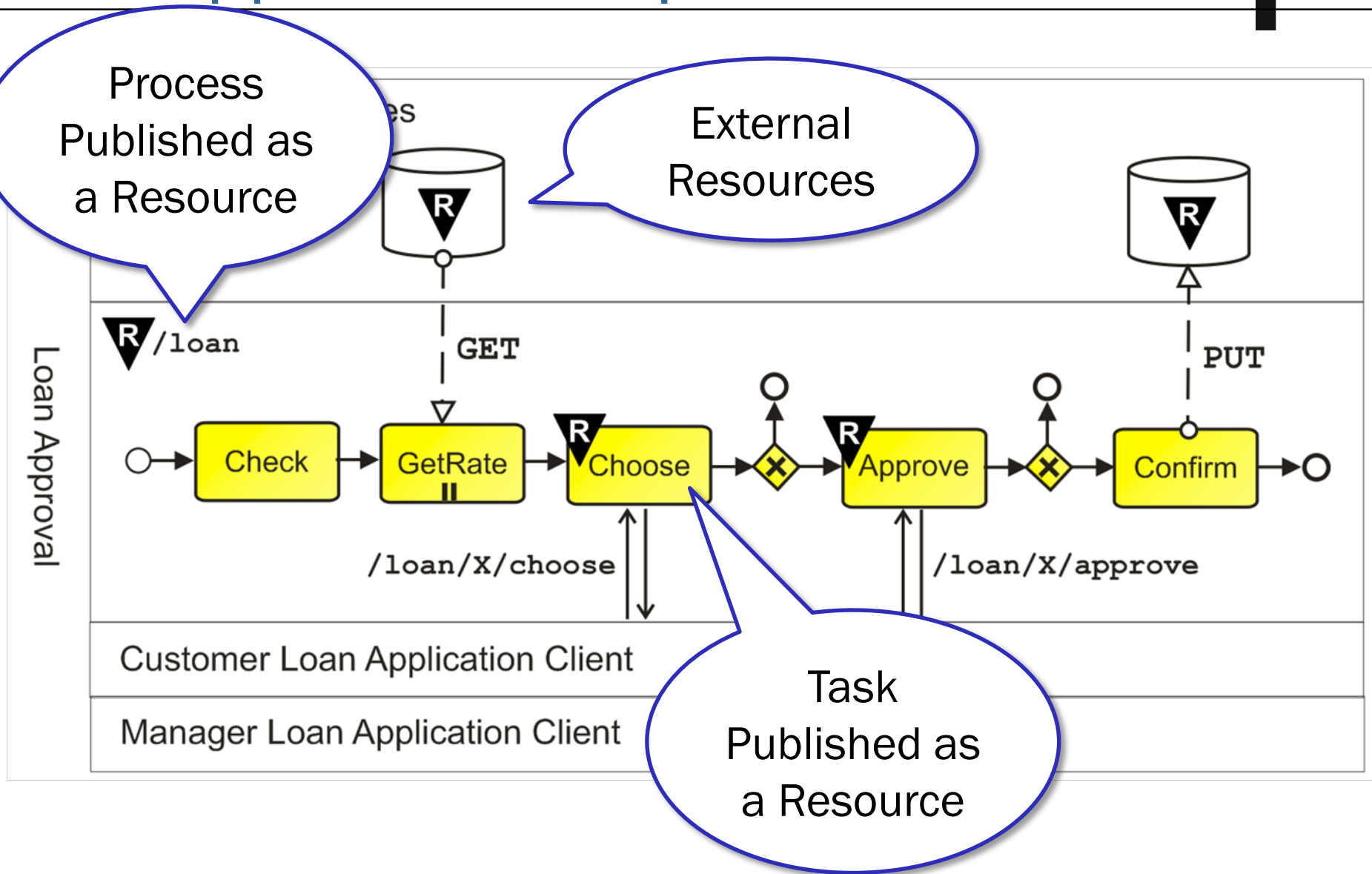
ContentType:

application/atom+xml

Web feed representing  
the current state of the  
process instance  
(collection of task instances)



# Loan Approval Example



# Process as a Web Feed

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Loan Approval Process</title>
  <subtitle>Instance x</subtitle>
  <link href="http://rest.jopera.org/loan/x" rel="self" />
  <link href="http://rest.jopera.org/loan" rel="template" />
  <link href="http://pubsubhubbub.appspot.com/" rel="hub" />
  <id>http://rest.jopera.org/loan/x</id>
  <updated>2011-06-10T11:11:30Z</updated>
  <author><name>Cesare Pautasso</name><email>cp@jopera.org</email></author>
  <entry>
    <title>Choose Task (Ready)</title>
    <link href="http://rest.jopera.org/loan/x/choose" />
    <id>http://rest.jopera.org/loan/x/choose</id>
    <updated>2011-06-10T11:12:20Z</updated>
    <summary>State: ready</summary>
  </entry>
  <entry>
    <title>Approve Task (waiting)</title>
    <link href="http://rest.jopera.org/loan/x/approve" />
    <id>http://rest.jopera.org/loan/x/approve</id>
    <updated>2011-06-10T11:11:30Z</updated>
    <summary>State: waiting</summary>
  </entry>
</feed>
```



# BPM

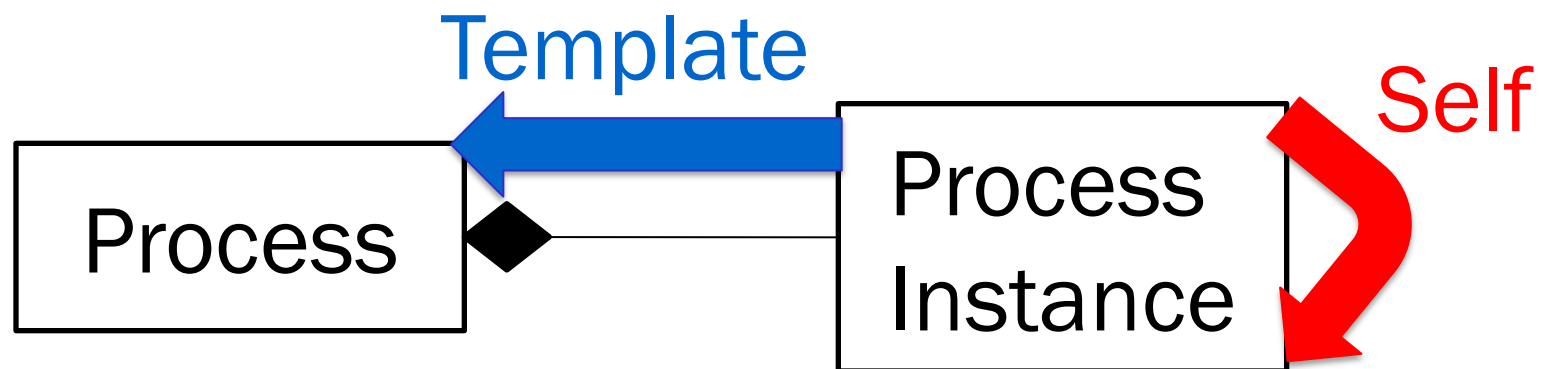
- Process Instance
- Task
- Process User
- Task State
- Task Timestamp
- Task Instance URI

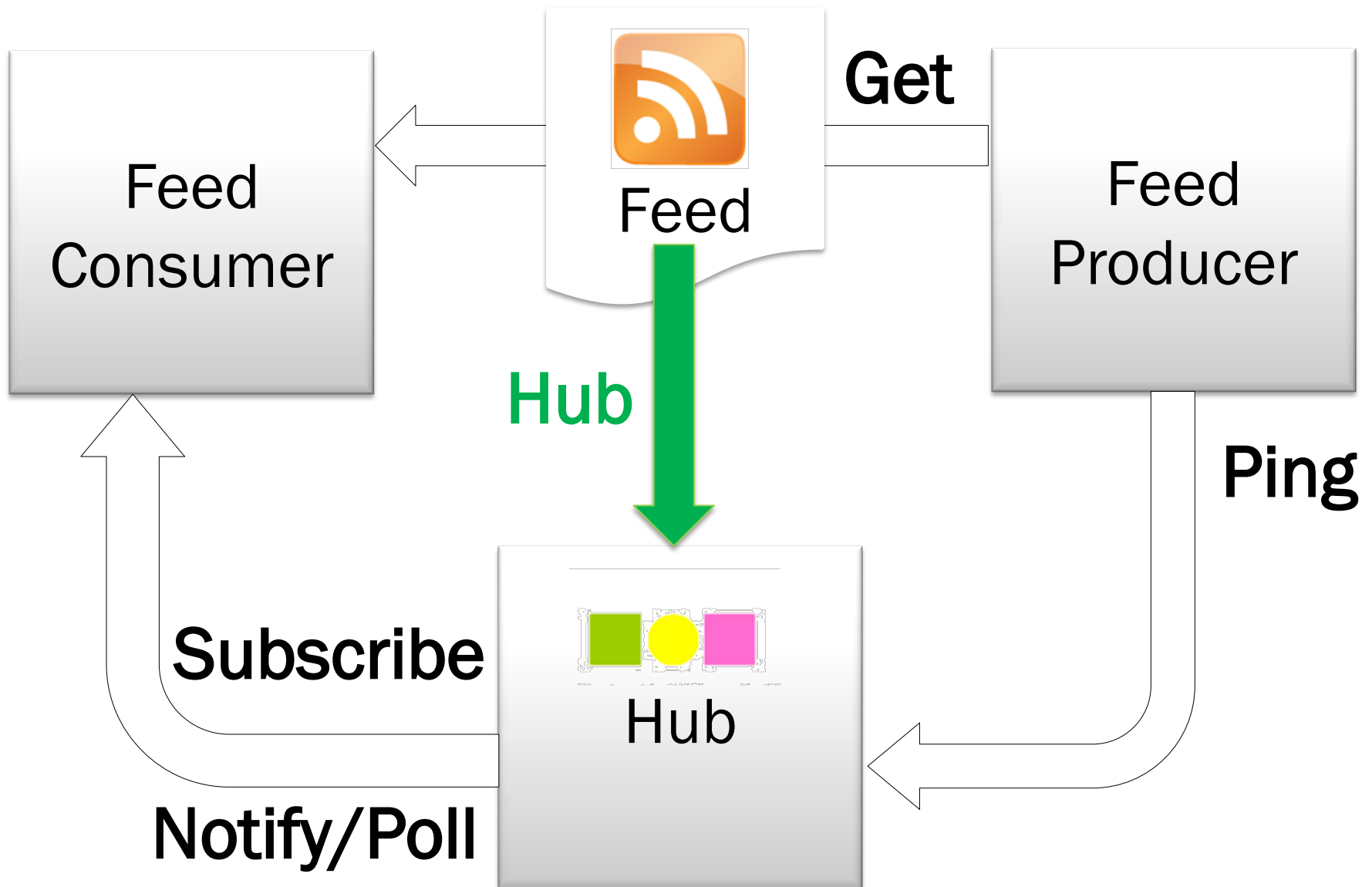
# Feed

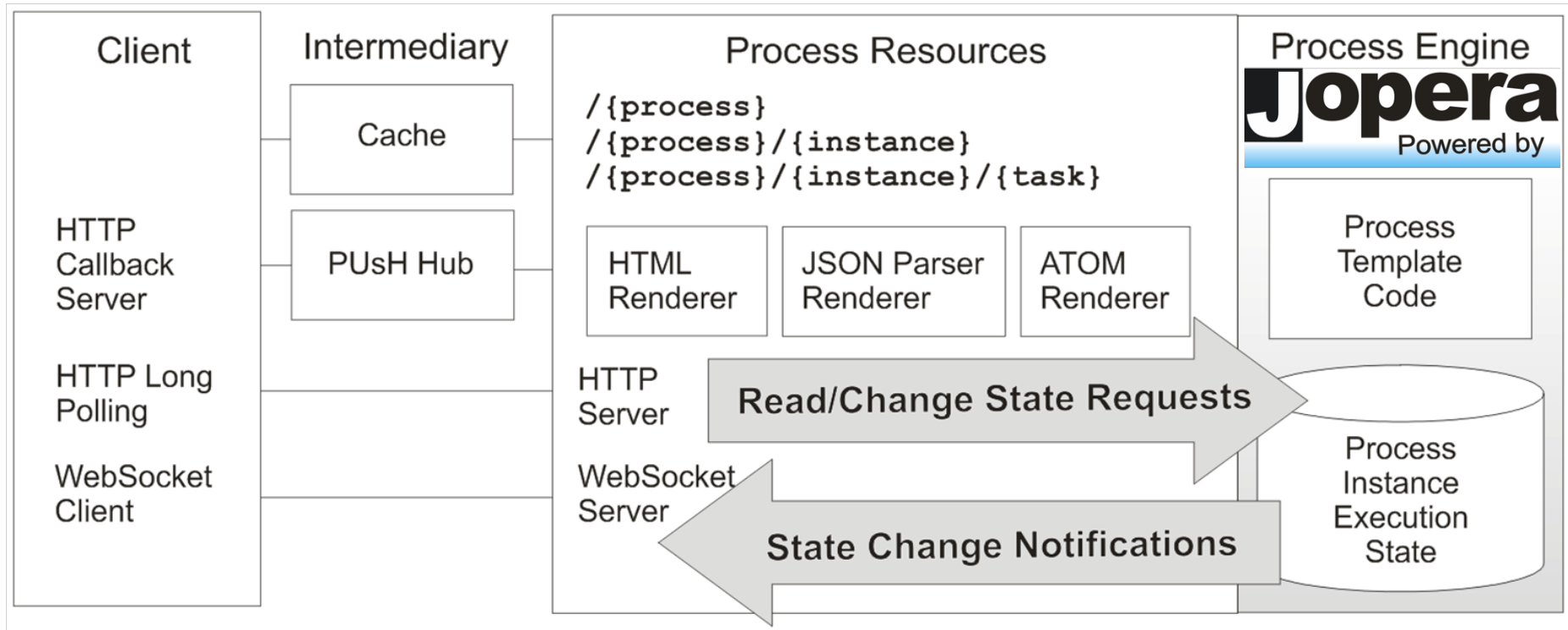
- Feed
- Feed Entry
- Feed Author
- Summary
- Updated
- Link

# Link Relations

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Loan Approval Process</title>
  <subtitle>Instance x</subtitle>
  <link href="http://rest.jopera.org/loan/x" rel="self"/>
  <link href="http://rest.jopera.org/loan" rel="template"/>
  <link href="http://pubsubhubbub.appspot.com/" rel="hub"/>
  <id>http://rest.jopera.org/loan/x</id>
</feed>
```







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

- Thanks to hypermedia, URIs and the HTTP uniform interface, REST resources are a very good abstraction to publish executable business processes on the Web
- RESTful HTTP is good enough to interact without any extension with process execution engines to drive the execution of process and task instances and to deliver notifications
- The state of a process instance can be projected to be represented as a standard Web feed
- The PubSubHubbub protocol can be used as an optimization to scale the corresponding delivery of notification callbacks



- Roy Fielding, [Architectural Styles and the Design of Network-based Software Architectures](#), PhD Thesis, University of California, Irvine, 2000
- Leonard Richardson, Sam Ruby, **RESTful Web Services**, O'Reilly, May 2007
- Jim Webber, Savas Parastatidis, Ian Robinson, **REST in Practice: Hypermedia and Systems Architecture**, O'Reilly, 2010
- Subbu Allamaraju, **RESTful Web Services Cookbook: Solutions for Improving Scalability and Simplicity**, O'Reilly, 2010
- Stevan Tilkov, **HTTP und REST**, dpunkt Verlag, 2009, <http://rest-http.info/>
- Thomas Erl, Raj Balasubramanians, Cesare Pautasso, Benjamin Carlyle, **SOA with REST**, Prentice Hall, end of 2010
- Martin Fowler,  
**Richardson Maturity Model: steps toward the glory of REST**,  
<http://martinfowler.com/articles/richardsonMaturityModel.html>

- Cesare Pautasso, Olaf Zimmermann, Frank Leymann, [RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision](#), Proc. of the 17th International World Wide Web Conference ([WWW2008](#)), Beijing, China, April 2008.
- Cesare Pautasso and Erik Wilde. [Why is the Web Loosely Coupled? A Multi-Faceted Metric for Service Design](#), Proc of the 18th International World Wide Web Conference ([WWW2009](#)), Madrid, Spain, April 2009.
- Cesare Pautasso, [BPEL for REST](#), Proc. of the 6th International Conference on Business Process Management ([BPM 2008](#)), Milan, Italy, September 2008.
- Cesare Pautasso, [BPMN for REST](#), Proc. of the 3<sup>rd</sup> BPMN Workshop (BPMN 2011), Luzern, Switzerland, November 2011
- Cesare Pautasso, [RESTful Web Service Composition with JOpera](#), Proc. Of the International Conference on Software Composition (SC 2009), Zurich, Switzerland, July 2009.
- Cesare Pautasso, Gustavo Alonso: **From Web Service Composition to Megaprogramming** In: Proceedings of the 5th VLDB Workshop on Technologies for E-Services (TES-04), Toronto, Canada, August 2004.



Raj Balasubramanians,  
Benjamin Carlyle,  
Thomas Erl,  
Cesare Pautasso,  
**SOA with REST**,  
Prentice Hall, 2012



# 10<sup>th</sup> International Conference on Business Process Management (BPM 2012)

September 3-6 2012, Tallinn, Estonia

<http://bpm2012.ut.ee>



# ws://rest.2012

Third International Workshop on RESTful Design

16-20 April 2012, Lyon, France

<http://ws-rest.org/2012>