Web Services with Zope and Plone

NORTH AMERICA
PLONE SYMPOSIUM
2005 JULY
NEW ORLEANS
LOUISIANA
20 21 22

Object realms

CIGNEX

PART 1 -- Overview

What are Web Services?

- XML technology
- SOAP
- WSDL
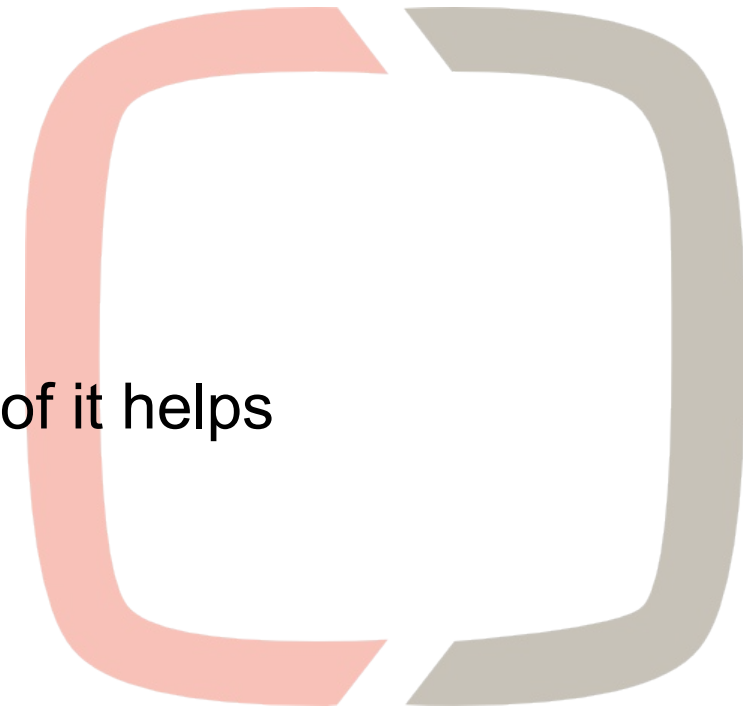- XML Schema
- UDDI

What is SOA?

- Service Oriented Architecture
  - Save vision as CORBA and DCOM implemented atop Web Services
  - Service Peer Model where applications are composed of loosely coupled, highly cohesive services based around interface rather than implementation.

How is this different from XML-RPC?

- Self describing*

- Deals well with complex data

- Extensible messaging protocol

  - via Headers and Actors

  - Web Services and SOA

* XSD is complex, using only a subset of it helps
create interoperable data.

What are Web Services and SOA?

- Service Oriented Architecture
- Self Contained
- Self Describing
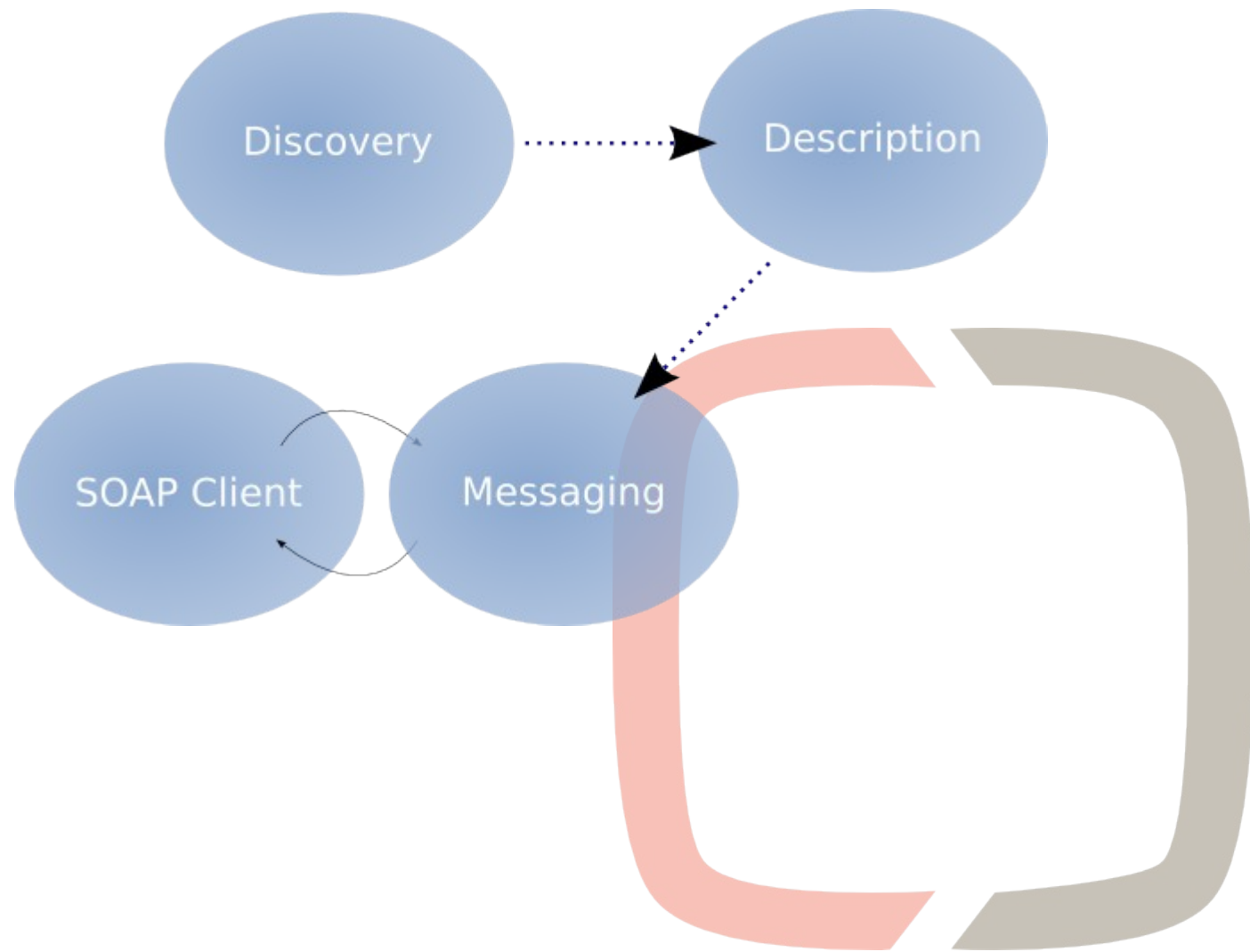- Decoupled Services
- Location Independent

XML Technology

- XPath
- XML Schema Definitions (XSD)
- *XSLT*
- *XQuery*

Web Services Standards (Generation 1)

- UDDI
- SOAP
- WSDL

Web Service Standards (Generation 2)

- Messaging Specifications
    - SOAP
    - SOAP + Attachments
    - WS-Addressing
    - WS-Transfer
- Description/Discovery
    - WSDL
    - WS-Policy
    - UDDI
    - WS-ResourceProperties

Web Service Standards (Generation 2 –– continued)

- QoS
  - WS-ReliableMessaging
  - Ws-ResourceLifetime
  - Transactions
    - WS-AtomicTransactions
    - WS-BusinessActivity
    - WS-Coordination
- Security
  - WS-Security
  - WS-Federation
  - WS-Trust
  - SAML
  - …

Web Service Standards (Generation 2 -- continued)

- Composition
  - BPEL4WS
  - WS-Notification
  - WS-Manageability
 - WS-DistributedManagement

And a few others. There are competing standards in many of these slots as well.

# Making Sense of it all

- WS-I Basic Profile
    - Defines a subset of these standards that should be interoperable
    - Clarifies ambiguity in specifications
    - Limits some of the flexibility of these many specifications to form a core offering that can be reasonably implemented on different platforms.
    - Its really the best we can hope to do right now.

PART 2 – Technology
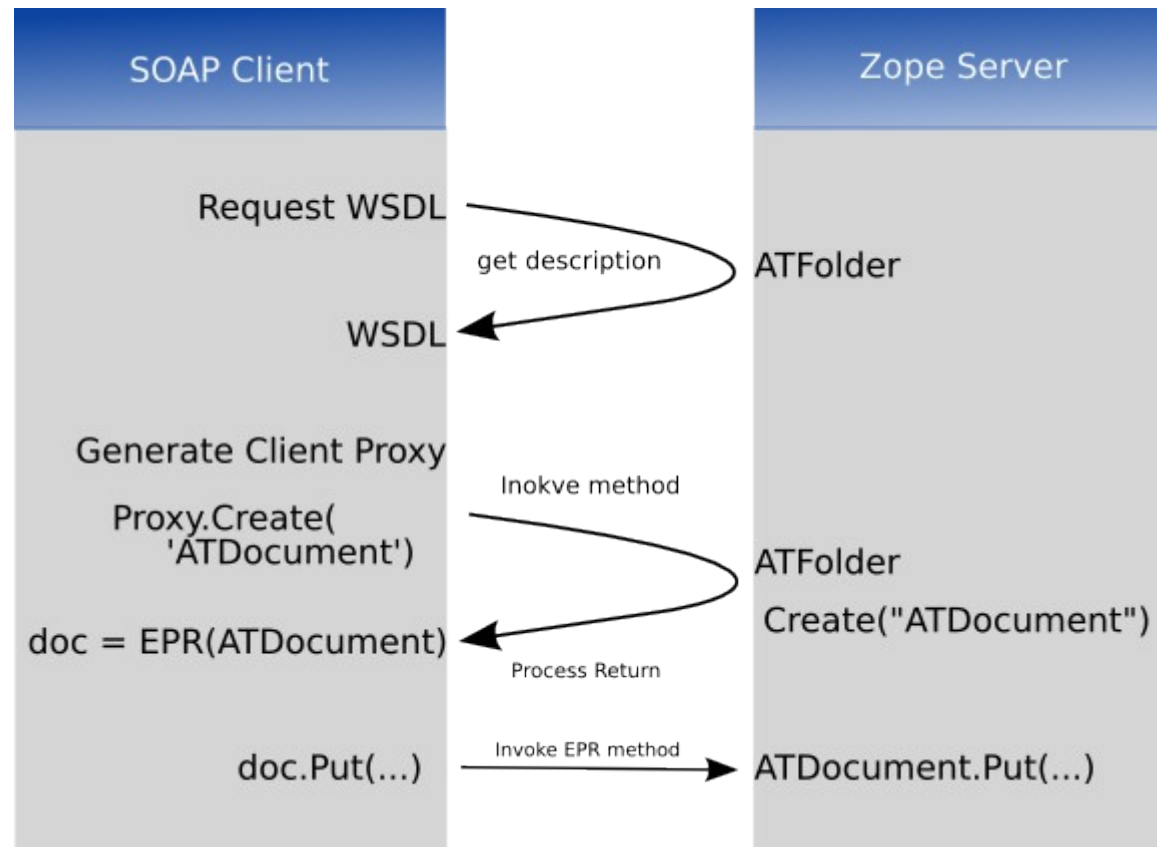
Messaging Styles

- RPC (SOAP encoding)

    - "Improved XML-RPC mode"

    - Encode Arguments/Response

- Document (document/literal)

    - "When you already have XML or don't have anything"

    - Send Document

    - Part of WS-I Basic Profile
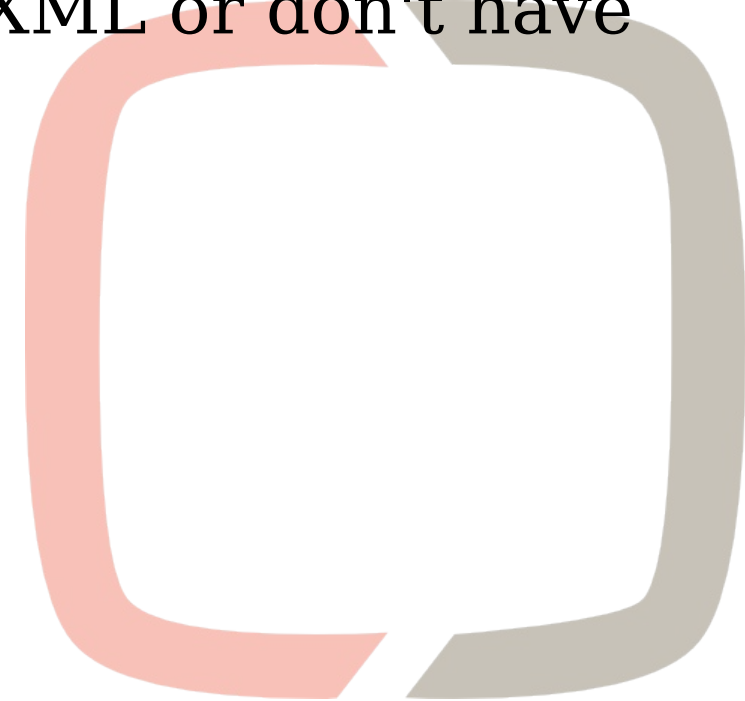
Publishing Models

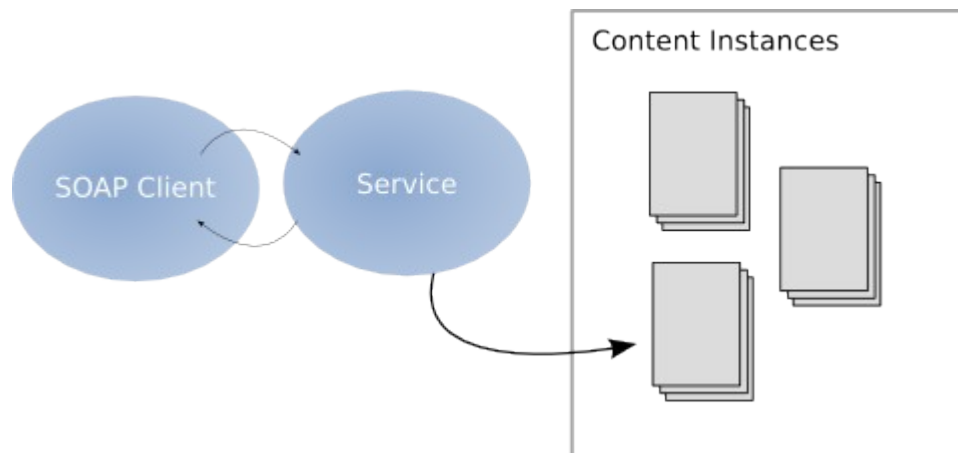- Service Publishing
- Object Oriented

Two slightly different ideas about Web Services.
The model choice here has real implications in the
flexibility and options you have moving forward.
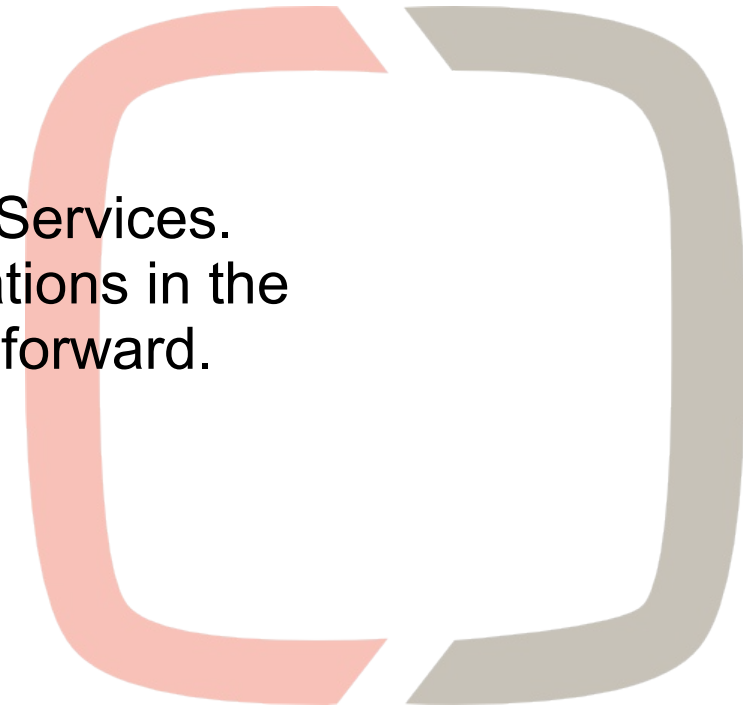
Service Oriented Publishing

- A service that deals through qualification with many different objects on their behalf. This model is quite common.
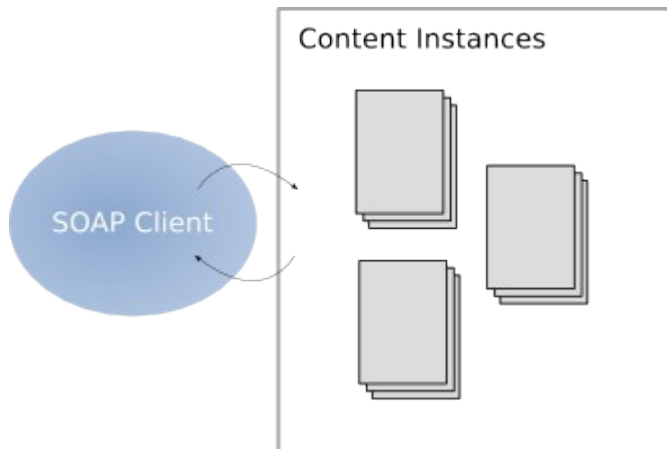
- The service exports the interface

```
application/serviceA.invoke(id="/site/objects/a")
```

Object Oriented Publishing

- Each system object is exposed with its own services and interfaces exposed on it. This is Zope's common model for publishing.
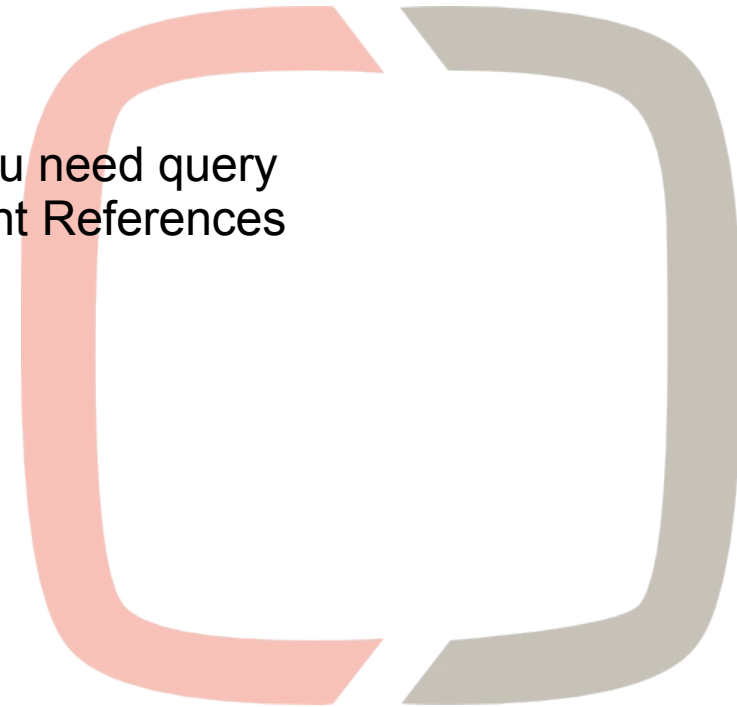
- The object exposes its interface

`application/documentA/Get()`

Implications

- Discoverablity

- Instance level uniqueness

In reality you need a bit of both and you need query
services with WS-Addressing End Point References

WS–Addressing and EPR

- Basis of most modern WS Standards
- WS-A defines how to target and deliver messages between services
- Implies a basic resource model and lifecycle
- Defines End Point References (EPR)
    - pointers back to other objects with enough embedded information that they should be invokable
    - Middleware can convert an EPR into a full object proxy

PART 3 Mechanics

```python
security.declareProtected(CMFCorePermissions.View, 'Get')
@wsdl(None, typecode)
def Get(self):
    """return self via SOAP"""
    return self
```

@WSDL

`@wsdl(expects, returns)`

Python2.4 decorator (callable in Python2.3 directly)

Annotates the method with typecode* information used in type marshaling and SOAP publishing

Archetypes objects can automatically provide their own typecodes.

```
@wsdl(None, typecode)
def Get(self):
    "Publish 'self' to SOAP"
    return self
```

* see ZSI.TC.* for more information on typecodes

auto_wsdl

```
auto_wsdl(method, expects, returns, **kwargs)
```

Not a descriptor.

Publish a method into the WSDL with less domain knowledge about typecodes.

Convenient access to schema fields using just their id.

```
auto_wsdl(MyAT.getTitleAndID,
          expects=None,
          returns=('title', 'id'))
```
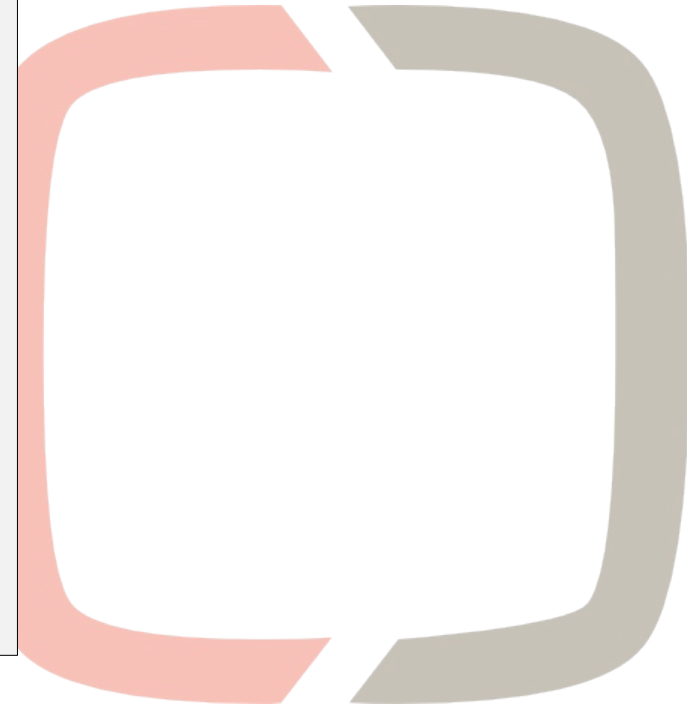
Client Examples

```
from SOAPpy import *                        SOAPpy

url = "http://localhost:8080/test/atd/WSDL"
proxy = WSDL.Proxy(url)
doc = proxy.Get()
print doc['title'], "from SOAPpy"
```

```
$ wget http://server/plone/document/WSDL     ZSI
$ wsdl2py -f WSDL

from ZSI import *
from ATDocument_services import *
import sys
from config import USER, PASS

loc = ATDocumentServiceLocator()
port = loc.getATDocumentPortType(**kwargs)

doc = port.Get(Empty())
print "Got", doc._title, "from",
      doc._url, "using ZSI."
```

Client Examples (continued)

```
$ wsdl http://server/test/doc/WSDL          MONO/.NET

using System;


class ATDocumentClient {

        public static void Main(String [] args) {

        ATDocumentService s = new ATDocumentService();
        s.Credentials = new System.Net.NetworkCredential(
                    "username", "password");

        // Get its state
        ATDocument doc = s.Get();

        Console.WriteLine("Got " + doc.title + " from "
                            + doc.url + " using Mono.");

        doc.title = doc.title + " updated";
        s.Put(doc);
        }
}
```

Cl

XMethods — Portal - Mozilla Firefox

File   Edit   View   Go   Bookmarks   Tools   Help

http://localhost:8080/test/xm/webservice_view

Daily   Weekly   Reference   Work   PMS   Suggest   Maps   CRM   blog   SOAPScope

small text   normal text   large text

Plone

search

home   news   members

admin   my folder   my preferences   undo   plone setup   log out

you are here: home » xmethods

navigation

contents   view   edit   properties
add new item ▾   state: visible ▾

Home
Members
Base Folder

XMethods

WSDL http://www.xmethods.net/wsdl/query.wsdl

getAllServiceNames (rpc/encoded)
getAllServiceSummaries (rpc/encoded)
getServiceDetail (rpc/encoded)
getServiceNamesByPublisher (rpc/encoded)
getServiceSummariesByPublisher (rpc/encoded)

recent items

No items published or changed since your last log-in.

More…

« **July 2005** »

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Plone and its visual design is Copyright © 2000-2005 by Alexander Limi, Alan Runyan, Vidar Andersen.

PLONE POWERED

This site conforms to the following standards:

SECTION 508   W3C AA   W3C XHTML   W3C CSS   ANY BROWSER

Done   Adblock

File Edit View Go Bookmarks Tools Help

http://localhost:8080/test/xm/getAllServiceNames/view

📁 Daily 📁 Weekly 📁 Reference 📁 Work 📄 PMS Ⓖ Suggest Ⓖ Maps 🔴 CRM 📄 blog 📄 SOAPScope »

small text   normal text   large text

P l o n e

🔍 search

| home | news | members |

👤 admin • my folder • my preferences • undo • plone setup • log out

you are here: home » xmethods » getallservicenames

navigation

- 🔴 Home
- 📁 Members
- 📁 Base Folder

recent items

No items published or changed since your last log-in.

More...

getAllServiceNames

▲ Service
http://www.xmethods.net/wsdl/query.wsdl

Request

rpc/encoded
This webmethod takes no arguments

⁝ invoke

Response

Data | XML

Result
   id uuid:F53A2E51-2DA2-D7CF-C65E-BECF11A99A9A
   name Digital Signature Service

   id uuid:58F5DD0A-5190-C4B8-581A-58BB3CBA8598
   name Digital Signature Verify

   id uuid:11449A77-250F-A7B0-B722-EC13BA3E4214
   name aql.com

   id uuid:1AF499AB-7A1B-7476-BC5A-23BF1715CB77
   name Zip Codes

   id uuid:20A3767C-E3FF-95A1-726B-665963410131

« July 2005 »

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Done

Adblock

What Makes a good WS?

- Expose and manipulate Domain Objects along transactional boundries

  - Domain level scoping

  - Transactional semantics that maintain Domain integrity

- Idempotent methods and tools (utilities)

Part 4 Limitations and Options

- Currently limited by the availablity of Python SOAP tools.
    - No proper request/response cycle
    - Inablity to layer messaging standards around current impls.
- Limited support in Archetypes
    - Schemata should map to XML namespaces
- Requires Horizontal domain knowledge to use.
    - have to understand
        - ZSI (Typecodes)
        - A bit of SOAP

Future

- A new SOAP library.
  - Useful in Python, Zope2/3 and Plone
  - Usable request/response cycle
  - DOM based for specification layering
- Schemata should specify namespace
- Reduce domain knowledge needed to produce and consume these services