December 3-6, 2007, Santa Clara Marriott, Santa Clara, CA

WS-Management on Rails

Klaus Kämpf

MANAGEMENT DEVELOPERS CONFERENCE

Mission

Make development of

Management Applications

easy and fun

Overview

- Openwsman
- Introducing Ruby
- WS-Management with Ruby
- WS-CIM with Ruby
- Rails application

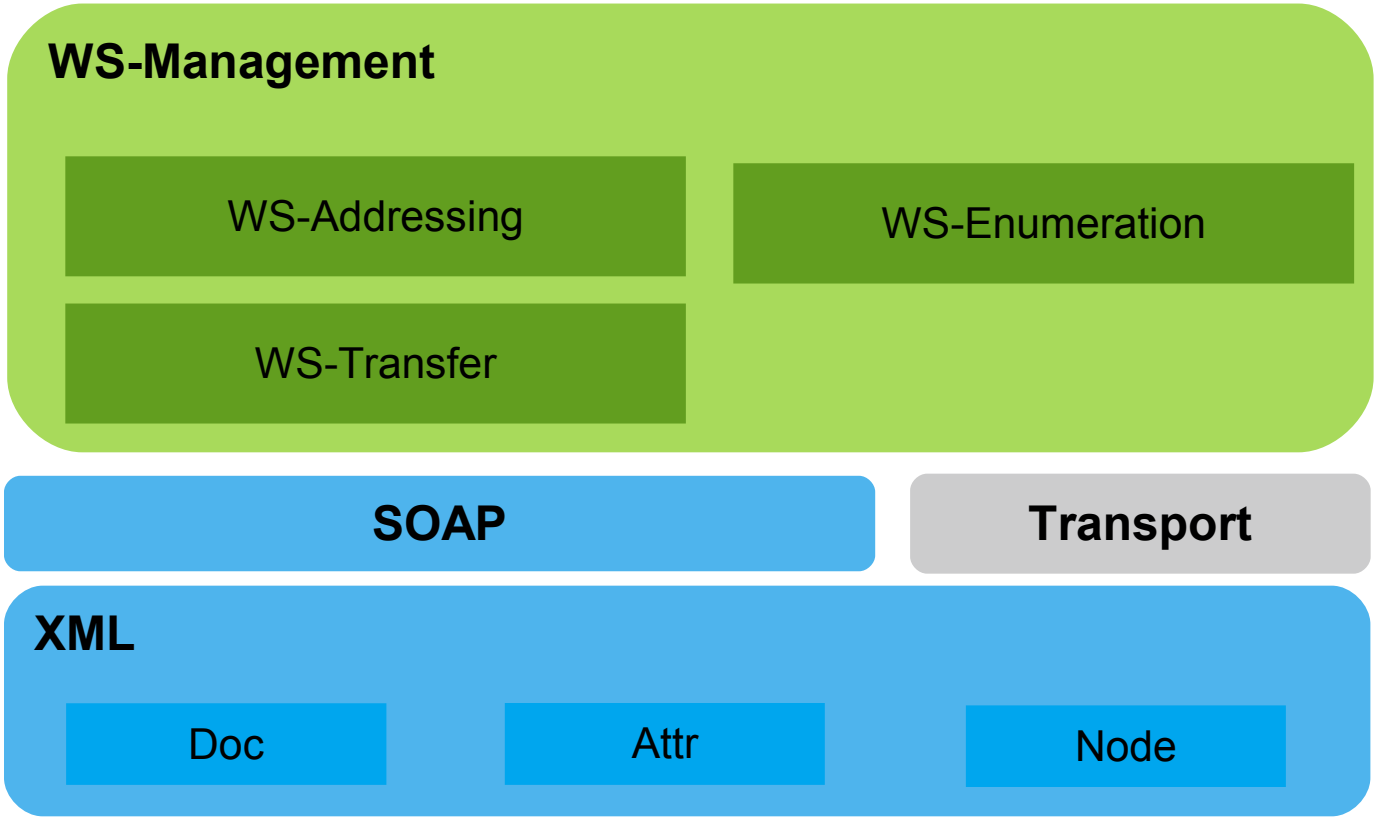
WS-Management

Openwsman

- WS-Management and WS-CIM
- Server and Client
- Open source, portable
- Native API in C
- C++ and Python bindings
- See http://www.openwsman.org

Openwsman Client Side

Openwsman Client API



MANAGEMENT DEVELOPERS CONFERENCE

Introducing Ruby

Ruby is a

- **dynamic**
- **object-oriented**

scripting language with

- a strong focus on simplicity and productivity
- an elegant, natural syntax
- roots in Lisp, Perl and Smalltalk
- easy ways to extend it

rwsman - WS-Management with Ruby

- XML Operations
- Transport Operations
- Create, Get, Put, Delete
- Enumerate
- Invoke
- Identify
- Eventing (future)

rwsman - Example

Stop the *Themes* service in Vista

```
1   require "wsman"
2
3   cimuri = "http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2"
4
5   client = WsMan::Client.new( "http://vista.microsoft.com:80/wsman" )
6
7   options = WsMan::ClientOption.new
8
9   options.selector_add( "Name", "Themes" )
10
11  client.invoke( cimuri+"/Win32_Service", "StopService", options )
```

WS-CIM

CIM_Service Instances Java

```
1   CIMClient cimClient = new CIMClient(
2           new CIMNameSpace("http://localhost", "root/cimv2"),
3           new UserPrincipal(""), new PasswordCredential(""));
4
5   CIMObjectPath op = new CIMObjectPath("CIM_Service");
6   Enumeration instances = cimClient.enumerateInstances(op);
7
8   while (instances.hasMoreElements()) {
9       CIMInstance service = (CIMInstance)instances.nextElement();
10
11      System.out.print(service.getProperty("Name").getValue().toString()
12        + " " +
13        service.getProperty("Started").getValue().toString()
14      );
15
16  }
```

CIM_Service Instances Perl

```
1    use Net::OpenWBEM::Client;
2
3    my $WBEM = Net::OpenWBEM::Client->session(host => "localhost");
4    my $ns = 'root/cimv2';
5    my $instanceEnum = $WBEM->enumInstancesE($ns, 'CIM_Service');
6
7    while ($instanceEnum->hasMoreElements)
8    {
9        my $ci = $instanceEnum->nextElement;
10       print $ci->getStringProperty('Name'), " ";
11       print $ci->getStringProperty('Started'), "\n";
12   }
13   $WBEM->close;
```

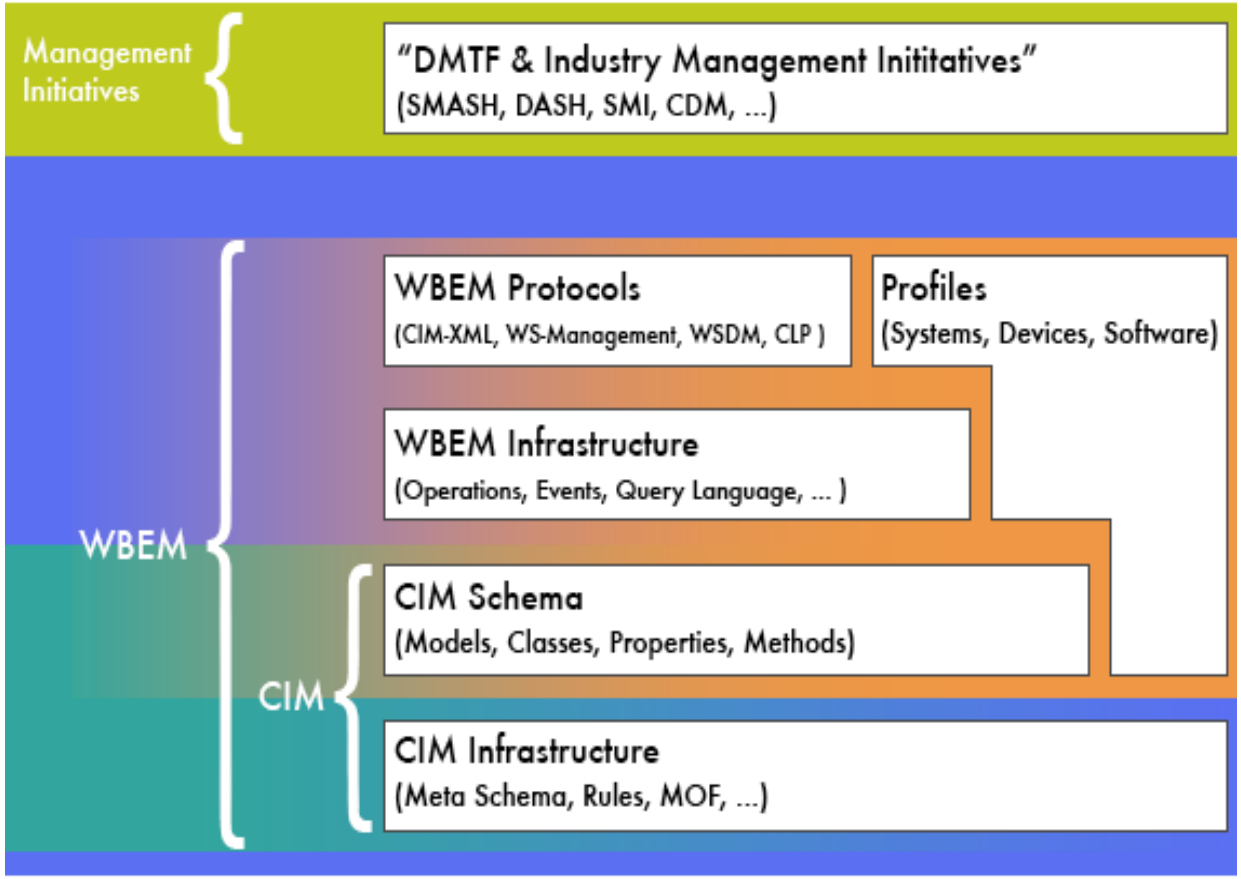
CIM_Service Instances Windows PowerShell

```
1   $colItems = get-wmiobject -class "CIM_Service" `
2   -namespace "root\cimv2" -computername "localhost"
3
4   foreach ($objItem in $colItems) {
5         write-host $objItem.Name, $objItem.Started
6   }
```

CIM_Service Instances Ruby

```
1    require "rwscim"
2
3    client = WsMan::Client.new( "http://localhost:8889/wsman”)
4
5    WsCim::Service.each( client ) { |service|
6      puts "#{service.Name}, #{service.Started}"
7    }
```

DMTF WBEM Stack



rcim

- Covers the CIM metamodel
  - Feature, Property, Method
  - Instance, Reference, Association
- Intrinsic operations

rcim

- Covers the CIM metamodel
  - Feature, Property, Method
  - Instance, Reference, Association
- Intrinsic operations

```
1    require "rwscim"
2    client = WsMan::Client.new( "http://localhost:8889/wsman”)
3    instance = WsCIM::Service.first
5    properties = instance.properties
6    keys = instance.keys
7    methods = instance.methods
8    references = instance.references
```

mofgen

- MOF -> C -> Ruby compiler
- Based on CIMPLE mof compiler
- Generates C extensions for Ruby
- CIM Instances as first-class Ruby Objects
- Obsolete with WS-Catalog

mofgen

- MOF -> C -> Ruby compiler
- Based on CIMPLE mof compiler
- Generates C extensions for Ruby
- CIM Instances as first-class Ruby Objects
- Obsolete with WS-Catalog

```
> cd cimple/schema/cim214
> ../../bin/mof2rwsman -I . CIM_Schema.mof -c CIM_Service
> ls cim_*
> cim_service.c  cim_service.h
```

rwscim

- WBEM Protocol stack
- Glues generated Ruby extensions

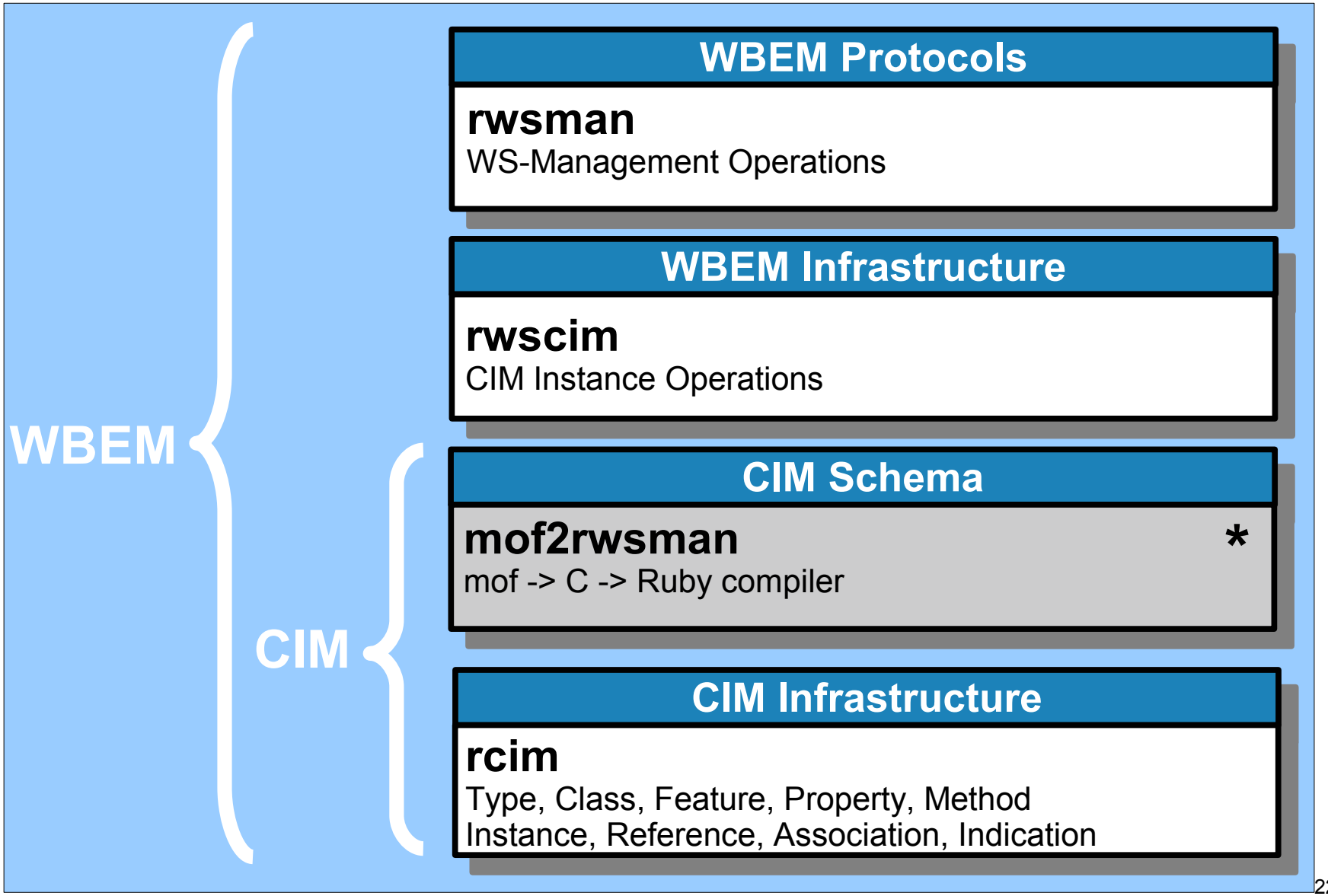
rwscim

- WBEM Protocol stack
- Glues generated Ruby extensions

```
1    require "rwscim"
2
3    client = WsMan::Client.new( "http://localhost:8889/wsman”)
4
5    WsCIM::Service.each( client ) { |service|
6      puts "#{service.Name}, #{service.Started}"
7    }
```

MANAGEMENT DEVELOPERS CONFERENCE

WBEM with Ruby



Web Management Application

MANAGEMENT DEVELOPERS CONFERENCE

Why Rails ?

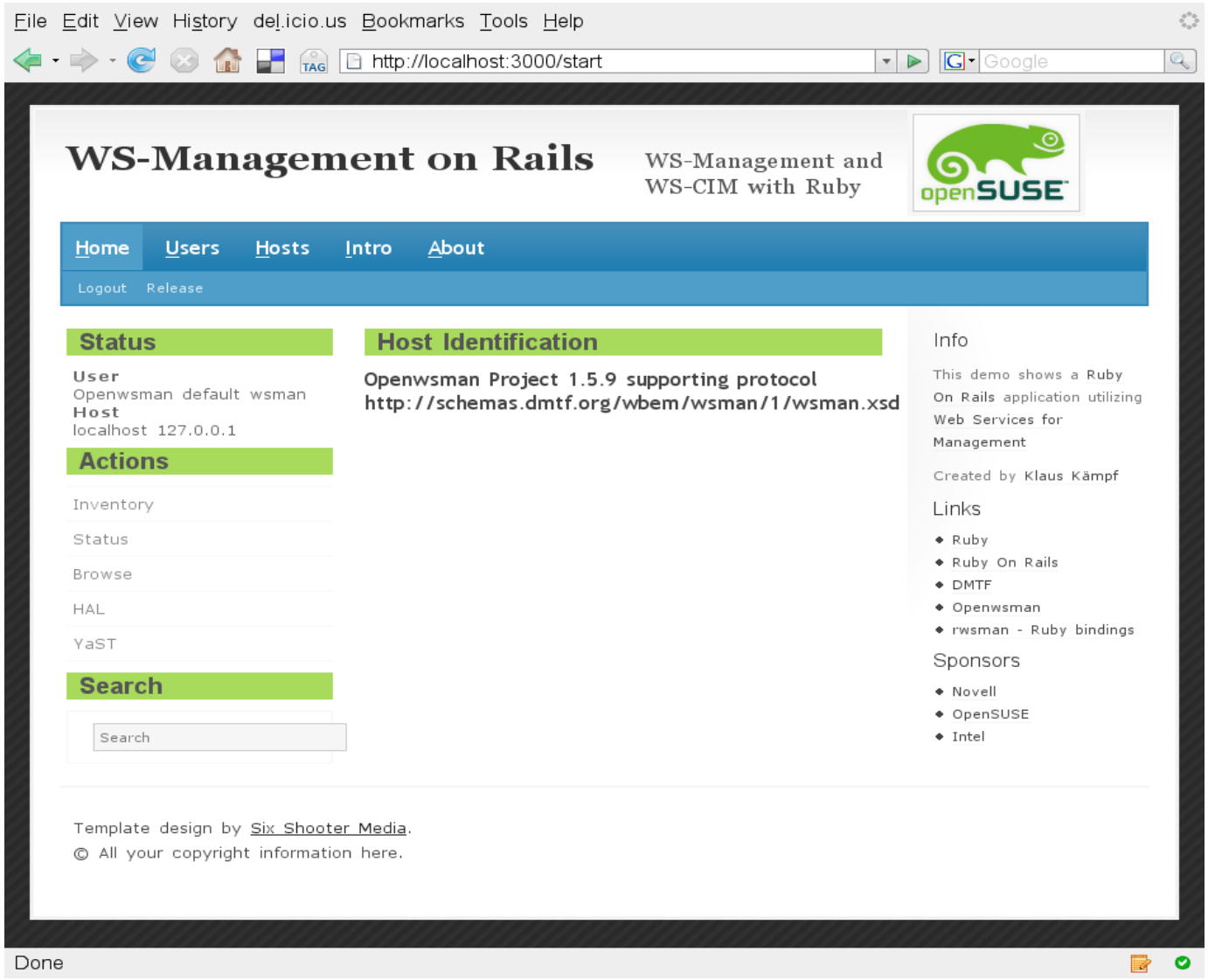
MANAGEMENT DEVELOPERS CONFERENCE

About Ruby On Rails

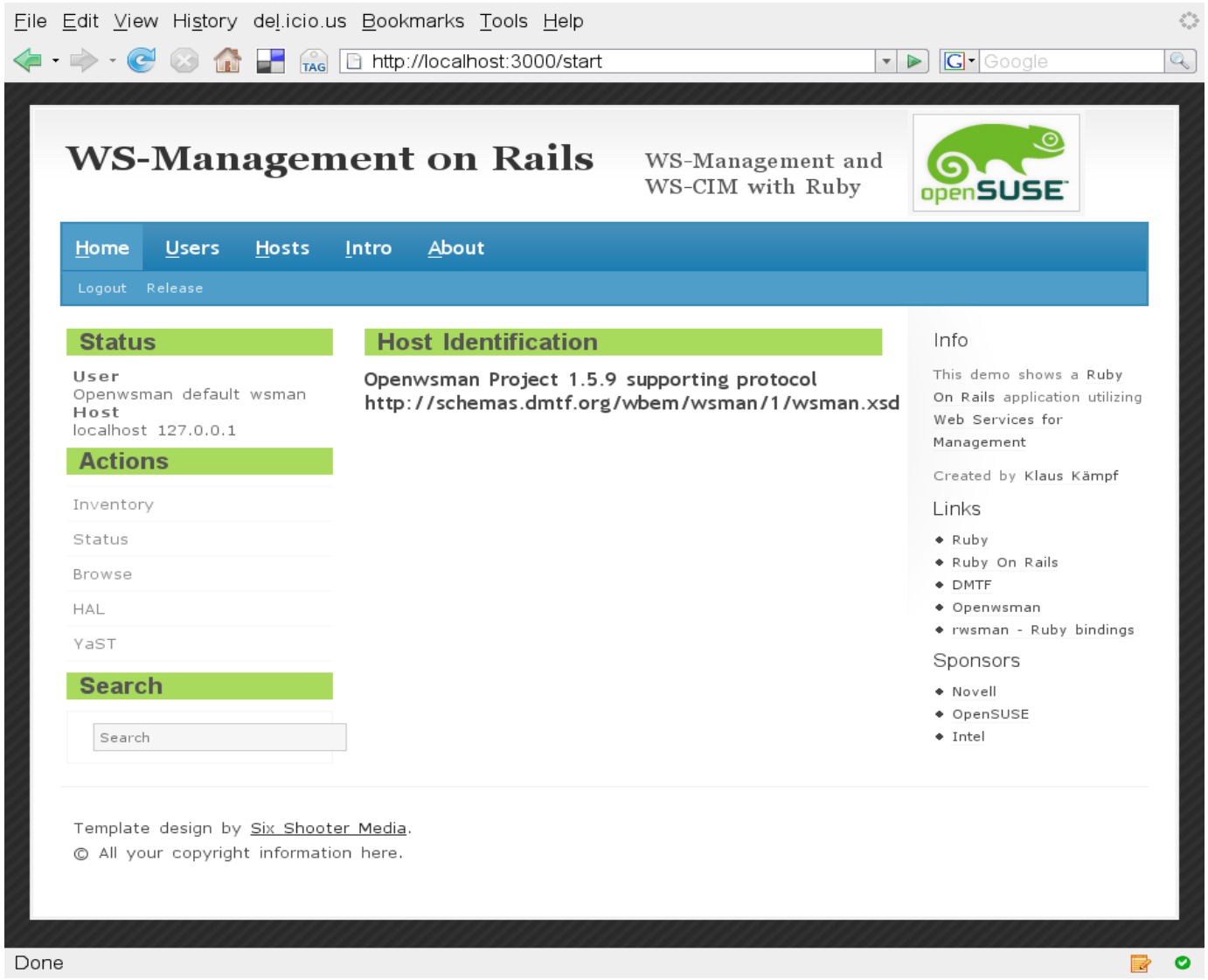
Web development that doesn't hurt

- Programmer happiness
- Productivity
- Convention over configuration
- MVC pattern
- No compilation
- Easy deployment and scaling
- Ajax

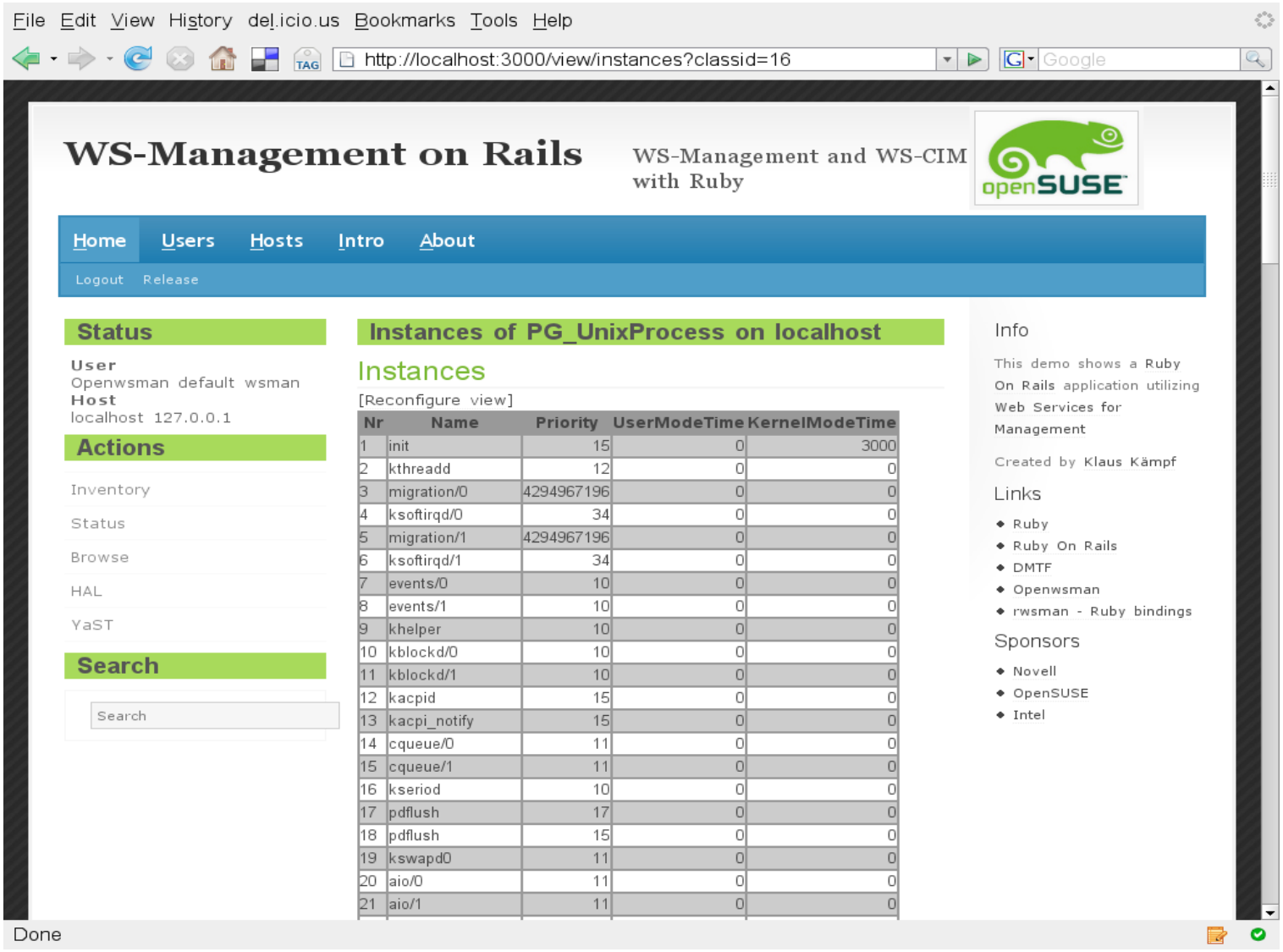
WS-Management on Rails



WS-Management on Rails



WS-Management on Rails



WS-Management on Rails

Where's the source ?

- Openwsman
  http://www.openwsman.org
- Ruby bindings for Openwsman
  http://rwsman.rubyforge.org
- Ruby On Rails
  http://www.rubyonrails.com
- RPM Packages
  http://en.opensuse.org/User:Kwk

Whats next ?

- On the fly MOF compilation
- Method invocation
- WS-Eventing
- WS-Catalog

  Dynamic creation of Ruby classes

That's all folks !

Thank you !

Mail to: kkaempf@suse.de

shell - SBLIM wbemcli

```
1  wbemcli ein http://localhost:5988/root/cimv2:CIM_Service |
2  while read instanceName; do
3      instance=`wbemcli gi http://$instanceName`
4      Name=`echo $instance
         | sed -n 's/^.*,Name="\([^"]*\)",.*$/\1/p'`
5      Started=`echo $instance
         | sed -n 's/^.*,Started=\([^,]*\),.*$/\1/p'`
6      echo "$Name $Started"
7  done
```