

Xen Summit
8 September, 2006

Xen Management API and Control Stack

Ewan Mellor
ewan@xensource.com

Xen Management Architecture

Cluster-wide Integration

The aim is to integrate Xen-based systems with enterprise-level cluster management solutions.

These solutions will use CIM as the integration interface. The Xen-CIM effort is of paramount importance.

Xen-CIM needs a stable interface into Xend.

Xen Management Architecture

Lightweight Management

We further aim to make it possible to manage Xen-based systems remotely, without enterprise-level solutions.

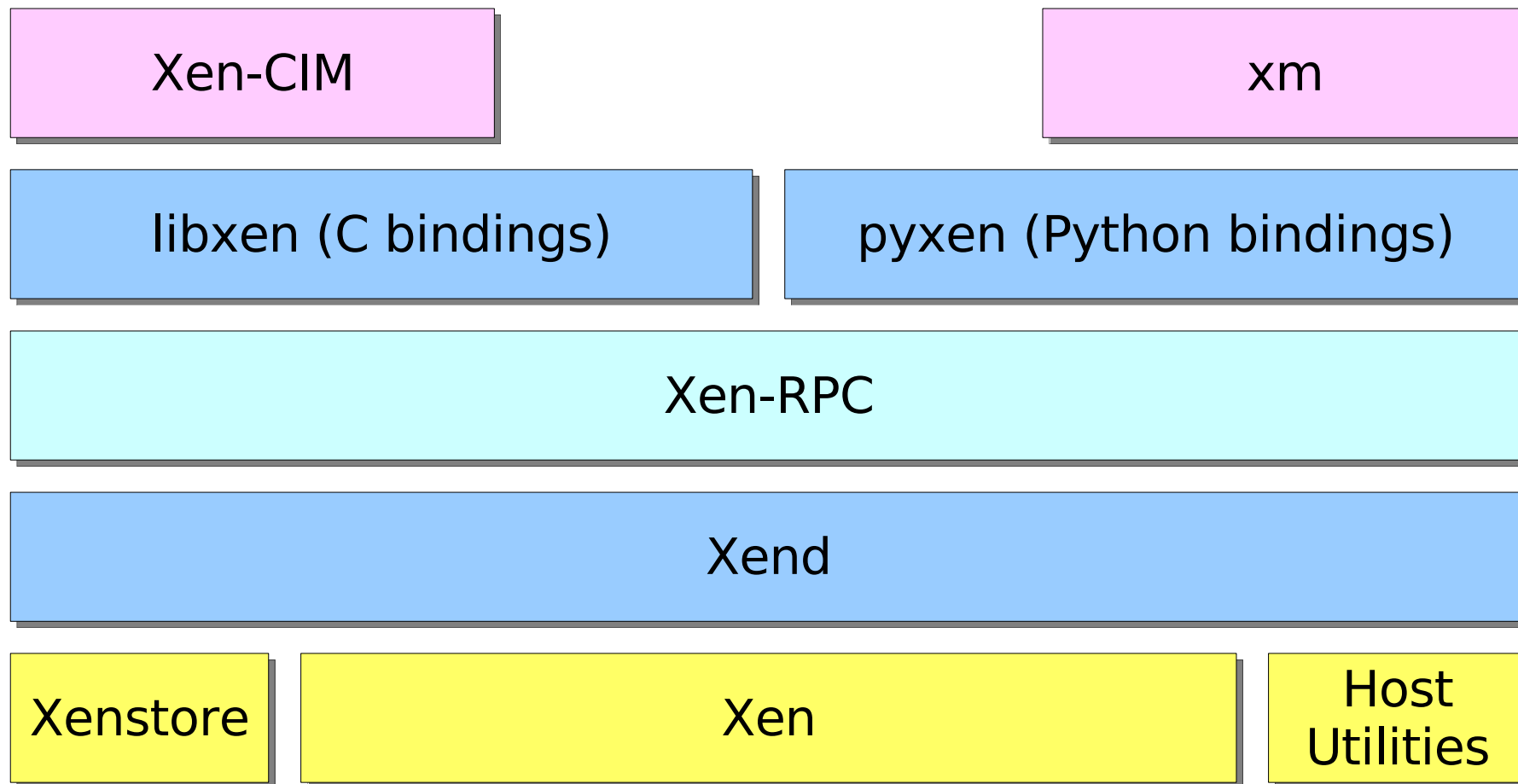
Build an ecosystem of third-party tools around Xen.

Keep the interfaces stable over the long term, allowing these tools to mature.

Remote Management

- Authentication
- Secure transports
- Management of non-running VMs
- Asynchronous notification mechanisms

Xen Management Architecture



Development Plan

- Develop within the xen-unstable tree following the Xen 3.0.3 fork
- Stabilise and release for Xen 3.0.4
 - Xend supporting new Xen-RPC
 - libxen (C bindings)
 - pyxen (Python bindings)

Future Guarantees

- Every client developed against Xen 3.0.4 will be supported in the long term at the wire level.
- Every client using libxen or pyxen will be supported within the same major version.

Migration Strategy

- Xen 3.0.4 will support the existing protocols as well as the new Xen-RPC.
 - FC6 will ship with tools using the legacy protocol.
 - OEMs and ISPs have in-house integration software that we don't wish to break.
- Integrators should be migrating to the new protocols as soon as possible.

Xen 3.0.3 Support

- RHEL 5 will be based on Xen 3.0.3
- Clients using Xen-RPC ought to be able to talk to RHEL 5

Xen 3.0.3 Adaptor

libxen / pyxen

Xen-RPC

Adaptor:
Features and messaging from Xend 3.0.4
Glue
Messaging from Xend 3.0.3

Legacy protocol

Xend 3.0.3

Example in C

Client application

libxen

GNOME's libxml2

A transport layer
e.g. nanohttp, curl

Example in C

```
xmlInitParser();
xen_init();
curl_global_init(CURL_GLOBAL_ALL);

xen_session *session =
    xen_session_login_with_password(call_func, NULL, username, password);

xen_vm vm;
if (!xen_vm_get_by_uuid(session, &vm, uuid)) {
    /* Error */
}

xen_vm_record *vm_record;
if (!xen_vm_get_record(session, &vm_record, vm)) {
    /* Error */
}

if (!xen_vm_start(session, vm)) {
    /* Error */
}
```

Example in C

```
static int call_func(const void *data, size_t len, void *user_handle,
                    void *result_handle, xen_result_func result_func) {

    CURL *curl = curl_easy_init();

    xen_comms comms = {
        .func = result_func,
        .handle = result_handle
    };
    curl_easy_setopt(curl, CURLOPT_URL, url);
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, &write_func);
    curl_easy_setopt(curl, CURLOPT_WRITEDATA, &comms);

    CURLcode result = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
    return result;
}

static size_t write_func(void *ptr, size_t size, size_t nmemb,
                        xen_comms *comms) {
    size_t n = size * nmemb;
    return comms->func(ptr, n, comms->handle) ? n : 0;
}
```

On-wire Example

```
xen_vm_record *vm_record;  
if (!xen_vm_get_record(session, &vm_record, vm)) {
```

```
<?xml version='1.0'?>  
<methodCall><methodName>Vm.get_record</methodName><params>  
  <param><value><string>12345678-abcd-ef90-123456789abc</string></value></param>  
</params></methodCall>
```

```
<?xml version='1.0'?>  
<methodResponse><params>  
  <param><value><struct>  
    <member><name>Status</name><value><string>Success</string></value></member>  
    <member><name>Value</name>  
      <value><struct>  
        <member><name>name_label</name>  
          <value><string>My VM</string></value></member>  
        <member><name>boot_method</name>  
          <value><string>kernel_external</string></value></member>
```

Xen Command Line Interface

Aims

Improve xm's configuration file handling:

- Migrate away from executable configuration files and SXP.

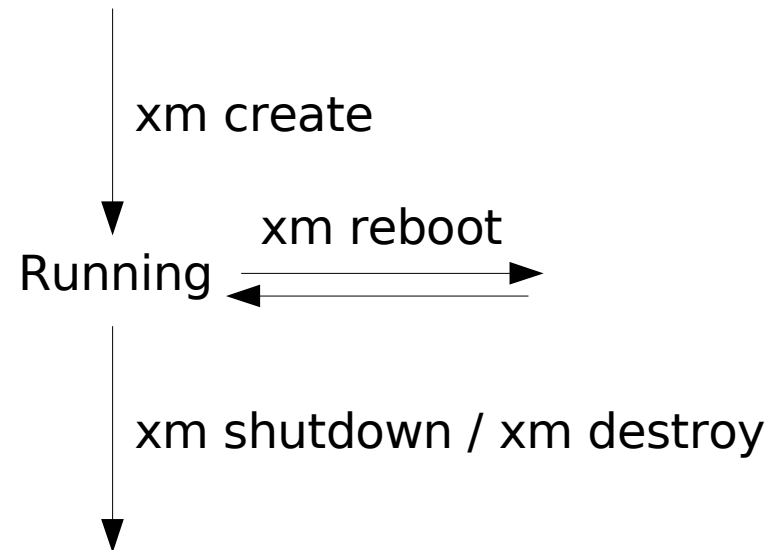
- Define a standard XML format.

- Support all the old formats in a backwards-compatible fashion.

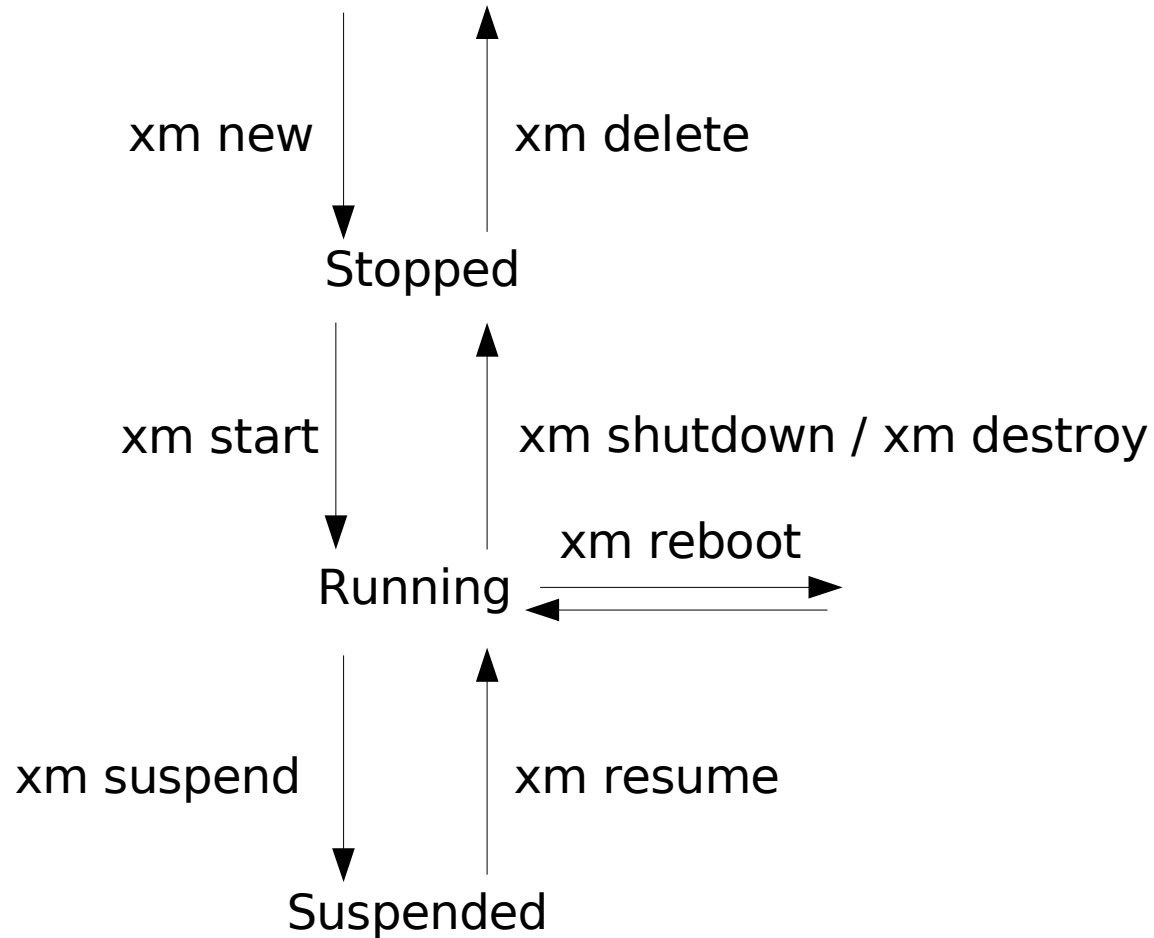
Expose new features from the API, in particular the authentication exchange.

Improve xm's syntax, tidy up the namespace.

xm new and xm create



xm new and xm create



xm new and xm create

- New XML-based configuration file
 - Closely tied to the Vm.create API call
- SXP-based and Python-based config files will be supported through a compat layer
- xm create preserved and equal to xm new followed by xm start
- 'transient' flag indicating that shutdown should trigger xm delete

Early Access

<http://wiki.xensource.com/xenwiki/XenApi>

<http://lists.xensource.com/cgi-bin/mailman/listinfo/xen-api>