Open Management Infrastructure (OMI)

A High-Performance Light-Weight CIM Server

Michael Brasher

Principal Software Development Engineer, Microsoft

mikbras@microsoft.com

Key Problems with Existing CIM Servers

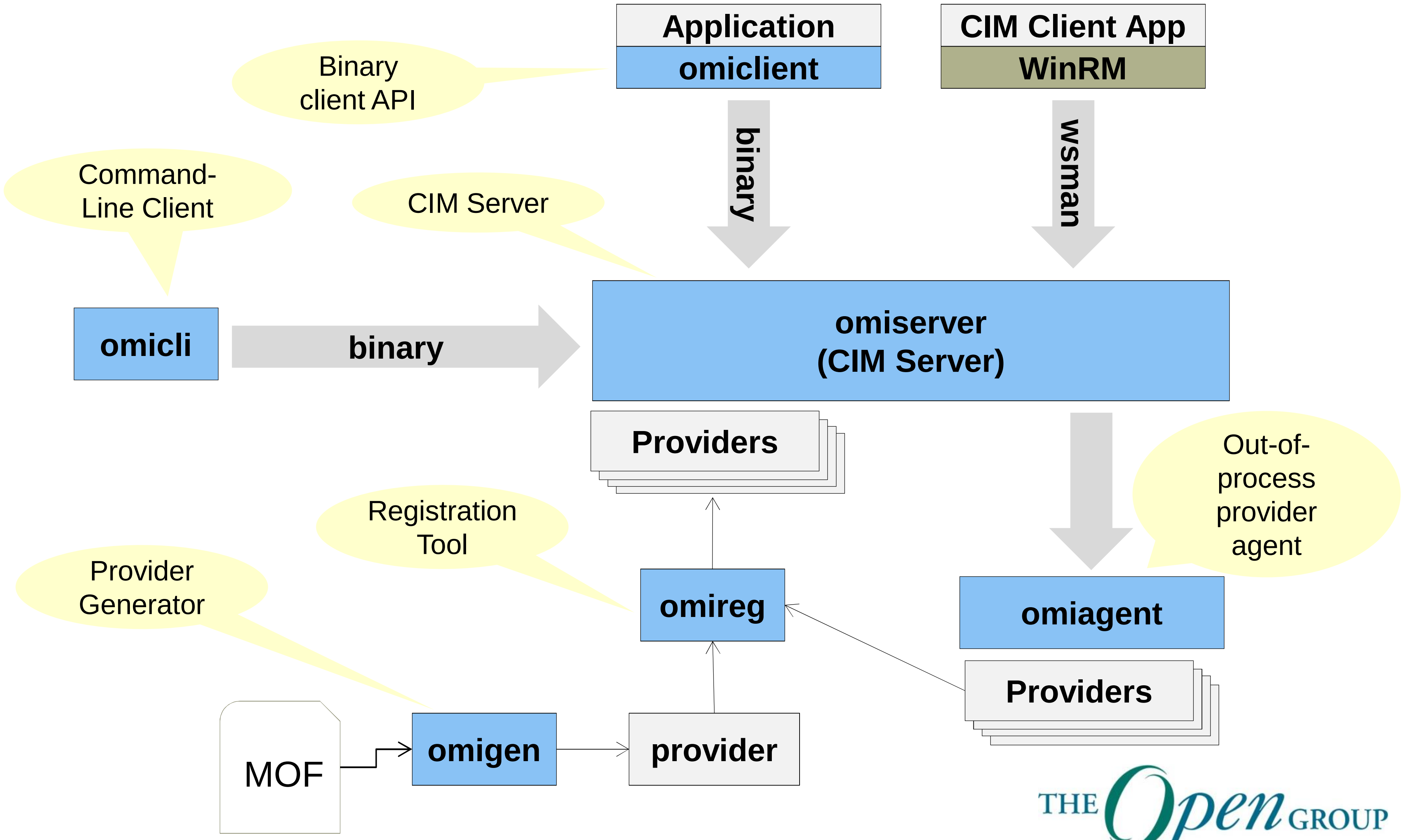
Problem	Consequence
Too heavy for small systems (embedded and mobility).	Vendors roll their own standard or proprietary solution.
Limited support for WS-Management.	Impedes WS-Management adoption and interoperability.
Providers are too hard to write.	Providers are very costly and unstable.
Provider interfaces are incompatible.	Vendors must write a provider for each CIM server (OpenPegasus/WMI).
Open-source CIM servers are hard to deploy.	Vendors customize the source distribution for their needs.

THE Open GROUP

What is OMI?

- A “second generation” CIM server
  - WS-Management is the primary protocol
  - Built for small systems (embedded & mobility)
  - Simplified (concrete) provider development
  - Easy to build and deploy
  - Portable (Unix, Linux)

Key OMI Elements



THE Open GROUP

Keeping it Small

- Server object size less than 250 kilobytes
- Server implemented entirely in C
- Provider interface is C
- Repository-less server
- Concrete provider classes yield less code
- Iterative size optimization
- Diskless operation

Provider Development Environment

- Next Generation Provider Interface
- Compatible with WMIv2 provider interface
- Generation of provider skeletons (omigen)
- Generation of concrete CIM classes structures (first-class objects)
- Registration through tool (omireg)

Dynamic vs. Concrete Provider Development

Dynamic (CMPI)

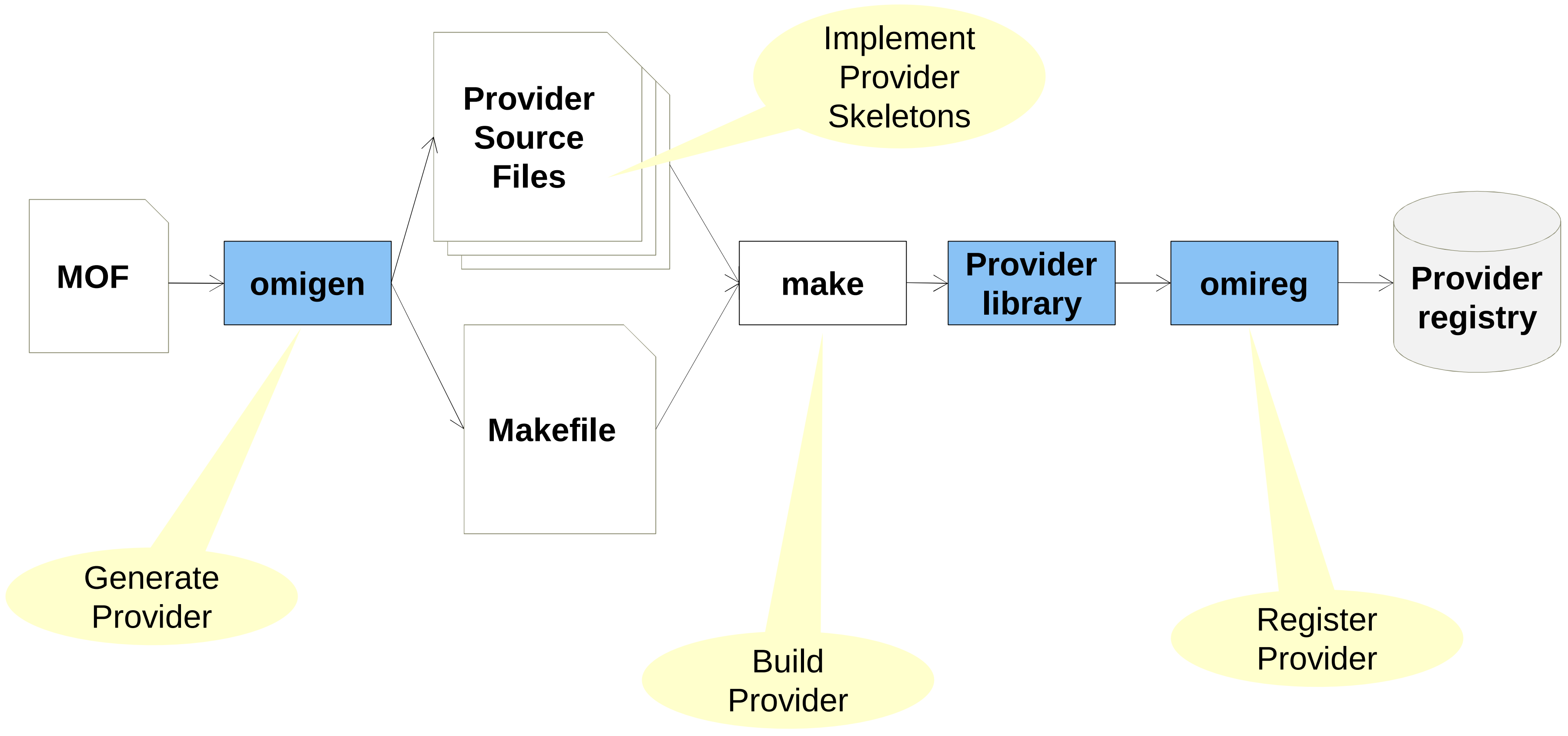
```
#incude <cmpimac.h>
#include <cmpidt.h>

CMPIInstance* frog;
CMPIStatus st;
CMPIValue v;
CMPIString* str;
…
if (!(frog = CMNewInstance(_cb, cop, &st)) || st.rc != CMPI_RC_OK)
    CMReturn(CMPI_RC_ERR_FAILED);

if (!(str = CMNewString(_cb, "1001", &st)) || st.rc != CMPI_RC_OK)
    CMReturn(CMPI_RC_ERR_FAILED);

v.string = str;
st = CMSetProperty(frog, "Key", &v, CMPI_string);

if (st.rc != CMPI_RC_OK)
    CMReturn(CMPI_RC_ERR_FAILED);

if (!(str = CMNewString(_cb, "Green", &st)) || st.rc != CMPI_RC_OK)
    CMReturn(CMPI_RC_ERR_FAILED);

v.string = str;
st = CMSetProperty(frog, "Color", &v, CMPI_string);

if (st.rc != CMPI_RC_OK)
    CMReturn(CMPI_RC_ERR_FAILED);

v.uint32 = 55;
st = CMSetProperty(frog, "Weight", &v, CMPI_uint32);

if (st.rc != CMPI_RC_OK)
    CMReturn(CMPI_RC_ERR_FAILED);

st = CMReturnInstance(result, frog);

if (st.rc != CMPI_RC_OK)
    CMReturn(CMPI_RC_ERR_FAILED);
```

Concrete (OMI)□

```
#include “Frog.h”
…
Frog frog;
Frog_Set_Key(&frog, "1001");
Frog_Set_Color(&frog, "Green");
Frog_Set_Weight(&frog, 55);
Frog_Post(&frog, context);
```

```
#include “Frog.h”
…
Frog frog;
Frog_Construct(&frog, context);
Frog_Set_Key(&frog, "1001");
Frog_Set_Color(&frog, "Green");
Frog_Set_Weight(&frog, 55);
Frog_Post(&frog, context);
```

Provider Development Process



THE Open GROUP

Security

- HTTPS (SSL)
- HTTP Basic Authentication
- Local Authentication
- PAM Authentication
- Out-of-process providers
  - Run as requestor
  - Run as server
  - Run as designated user

Repository

- No instance repository
- Immutable in-memory class repository (class information supplied by providers)
- Provider registration through flat files (rather than CIM instances)

Platforms

- Linux (Redhat & Suse)
- Solaris (x86 and SPARC)
- HPUX
- AIX
- Mac OS
- Windows (partial)

Build and Install

- GNU configure and build standards:
  - ./configure --prefix=/usr/local
  - make
  - make install
- Very configurable (./configure --help)
- GNU makefiles
- Build a distribution:
  - make dist
- Binary installer

THE Open GROUP

Key OMI Features

- WS-Management stack
- Local binary protocol
- In-process providers
- Out-of-process providers
- Process-based user security model
- HTTPS/SSL
- HTTP basic authentication
- PAM authentication
- WMIv2 providers
- Automated provider generation
- Automated provider registration
- ‘Repository-less’ server
- Small footprint
- Small image size
- WQL (WBEM Query Language)

Non-Features

- CMPI
- CIM-XML Protocol
- CQL
- Indications (events)
- Instance Repository