Jasper van Baten – AmsterCHEM
Bill Barrett – EPA
Michael Hlavinka – BR&E
Michel Pons – CO-LaN

STATUS OF COBIA

CO-LaN's CAPE-OPEN Middleware

MARSHALING & THREADING

CAPE-OPEN Annual Meeting 2021, October 27-28

PRESENTATION OUTLINE

- Marshaling & Threading
- Marshaling, short demo
- The COBIA threading models
- Mapping between COM and COBIA threading models

MARSHALING…

PME (Win, x64)

PMC (Win, x64, DLL)

MARSHALING…

PME (Win, x64)

PMC (Win, x64, DLL)

Thread

PMC (Win, x64, DLL)

MARSHALING…

PME (Win, x64)

PMC (Win, x64, DLL)

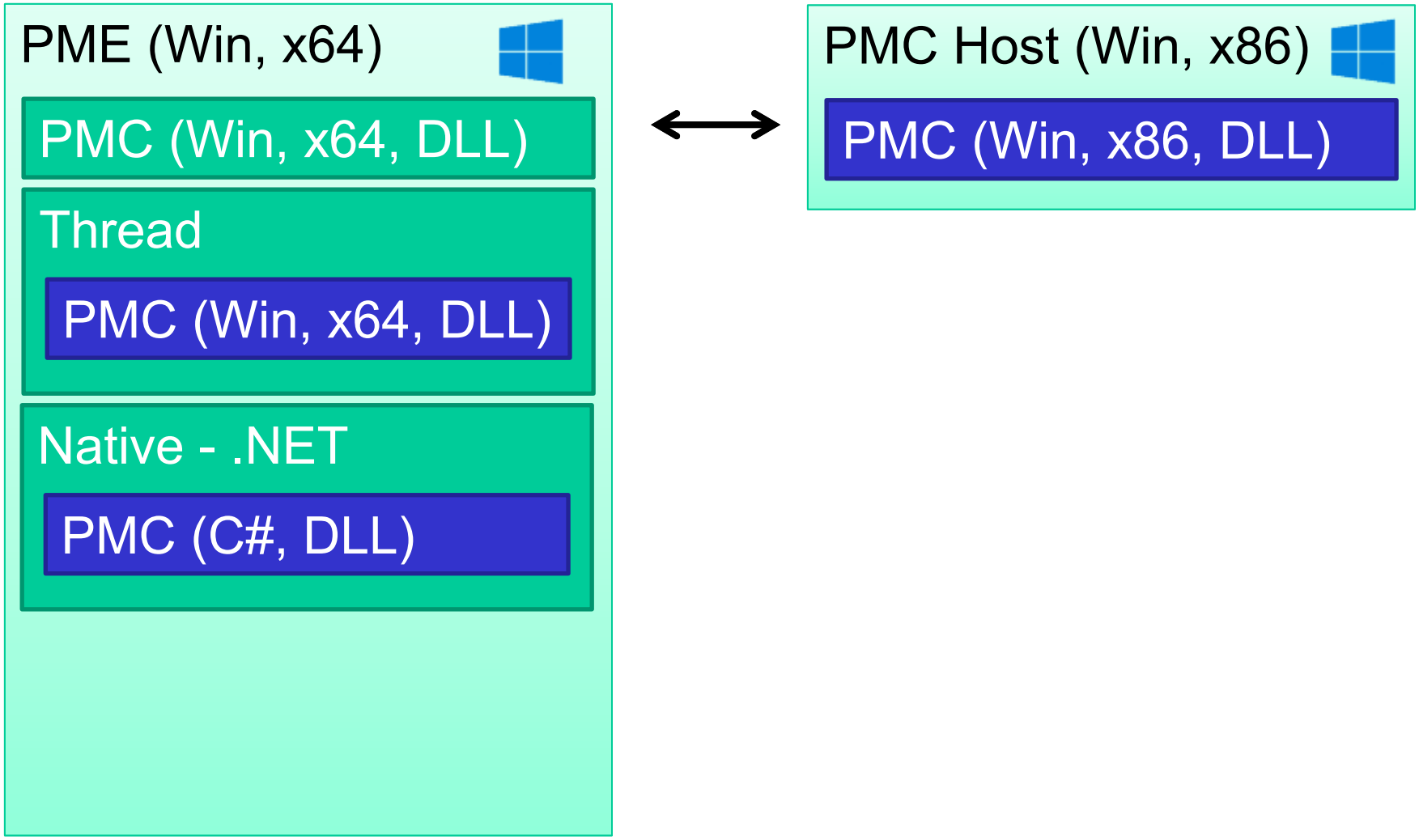
Thread

PMC (Win, x64, DLL)

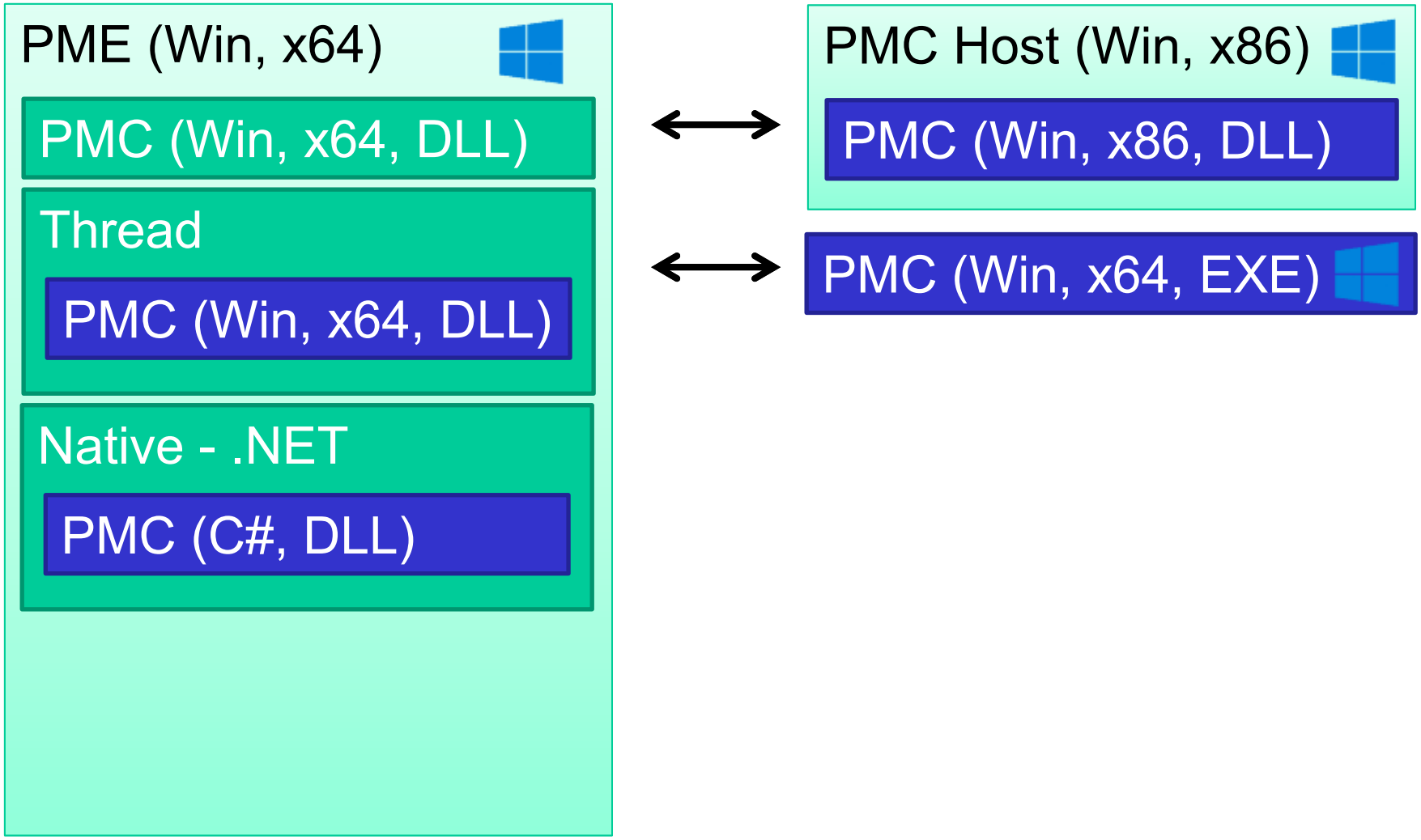
Native - .NET

PMC (C#, DLL)

MARSHALING…


PME (Win, x64)
PMC (Win, x64, DLL)
Thread
PMC (Win, x64, DLL)
Native - .NET
PMC (C#, DLL)
PMC Host (Win, x86)
PMC (Win, x86, DLL)

MARSHALING…


PME (Win, x64)
PMC (Win, x64, DLL)
Thread
PMC (Win, x64, DLL)
Native - .NET
PMC (C#, DLL)
PMC Host (Win, x86)
PMC (Win, x86, DLL)
PMC (Win, x64, EXE)

MARSHALING…

PME (Win, x64)

PMC (Win, x64, DLL)

Thread

PMC (Win, x64, DLL)

Native - .NET

PMC (C#, DLL)

PMC Host (Win, x86)

PMC (Win, x86, DLL)

PMC (Win, x64, EXE)

TCP/IP

Somewhere else:

PMC Host (Linux)

PMC (Linux, so)

MARSHALING…

PME (Win, x64)

PMC (Win, x64, DLL)

Thread

PMC (Win, x64, DLL)

Native - .NET

PMC (C#, DLL)

PMC Host (Win, x86)

PMC (Win, x86, DLL)

PMC (Win, x64, EXE)

TCP/IP

SSH

Somewhere else:

PMC Host (Linux)

PMC (Linux, so)

MARSHALING…

PME (Win, x64)

PMC (Win, x64, DLL)

Thread

PMC (Win, x64, DLL)

Native - .NET

PMC (C#, DLL)

Logging:

PMC (Win, x64, DLL)

PMC Host (Win, x86)

PMC (Win, x86, DLL)

PMC (Win, x64, EXE)

TCP/IP

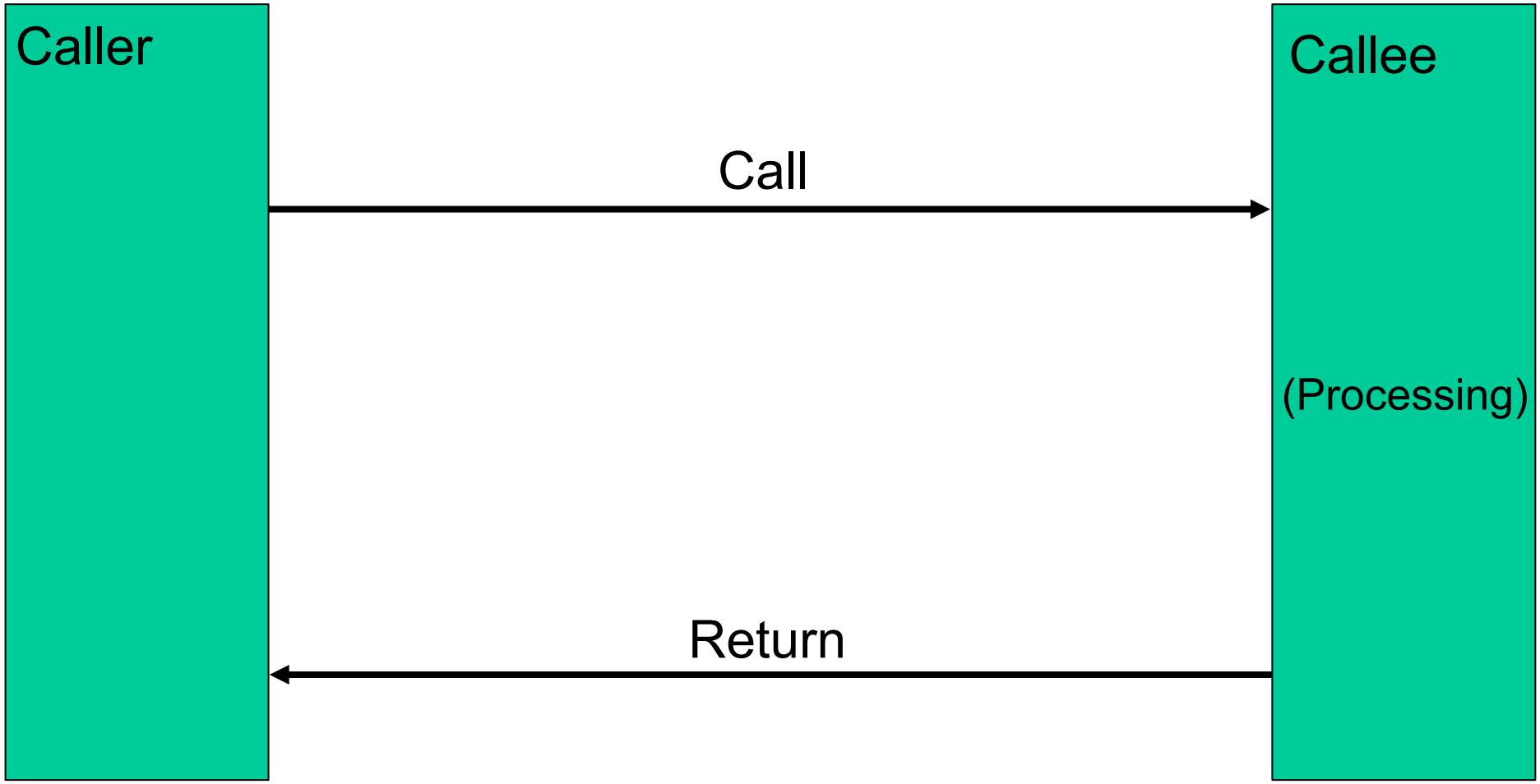
SSH

Somewhere else:

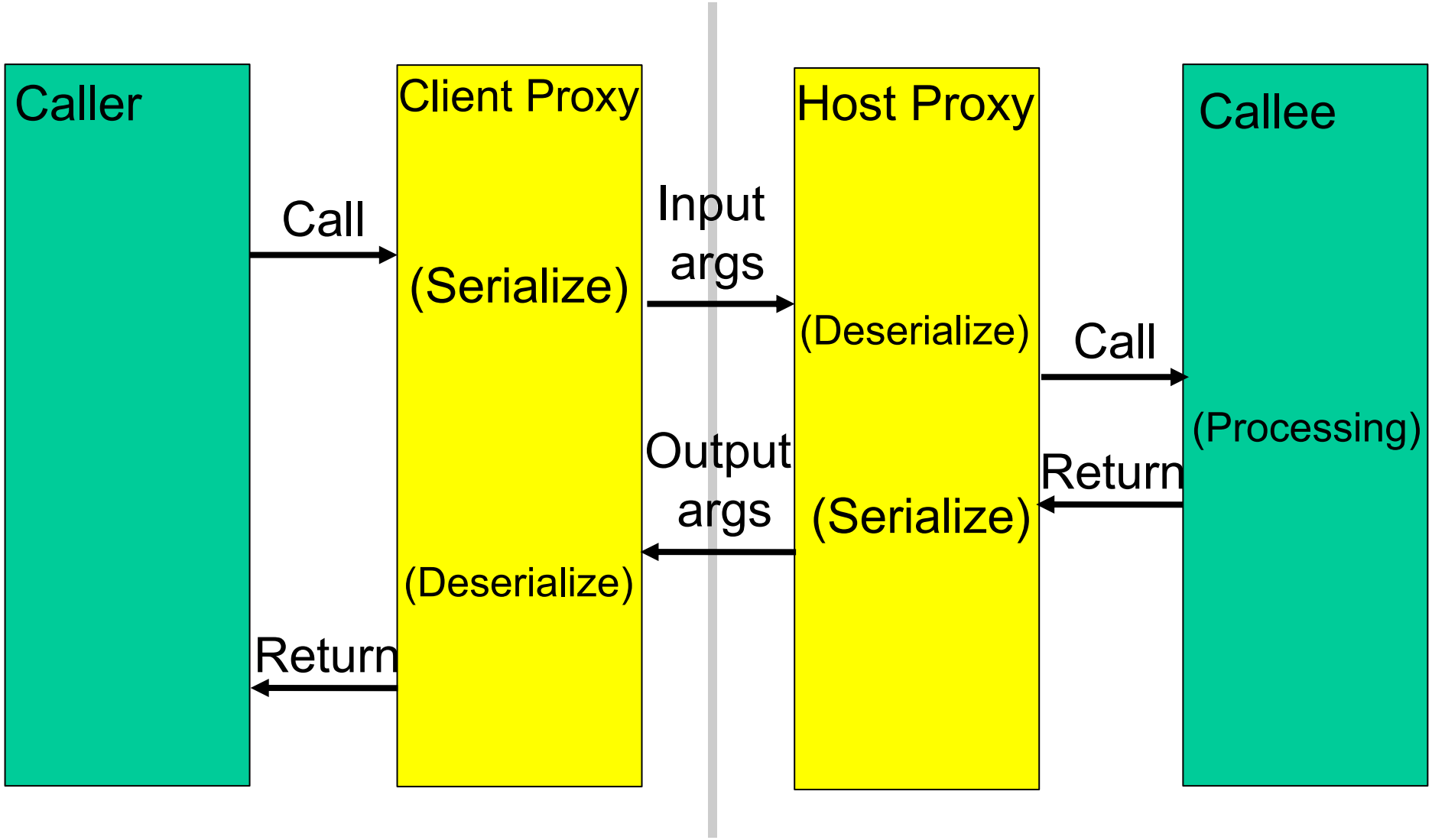
PMC Host (Linux)

PMC (Linux, so)

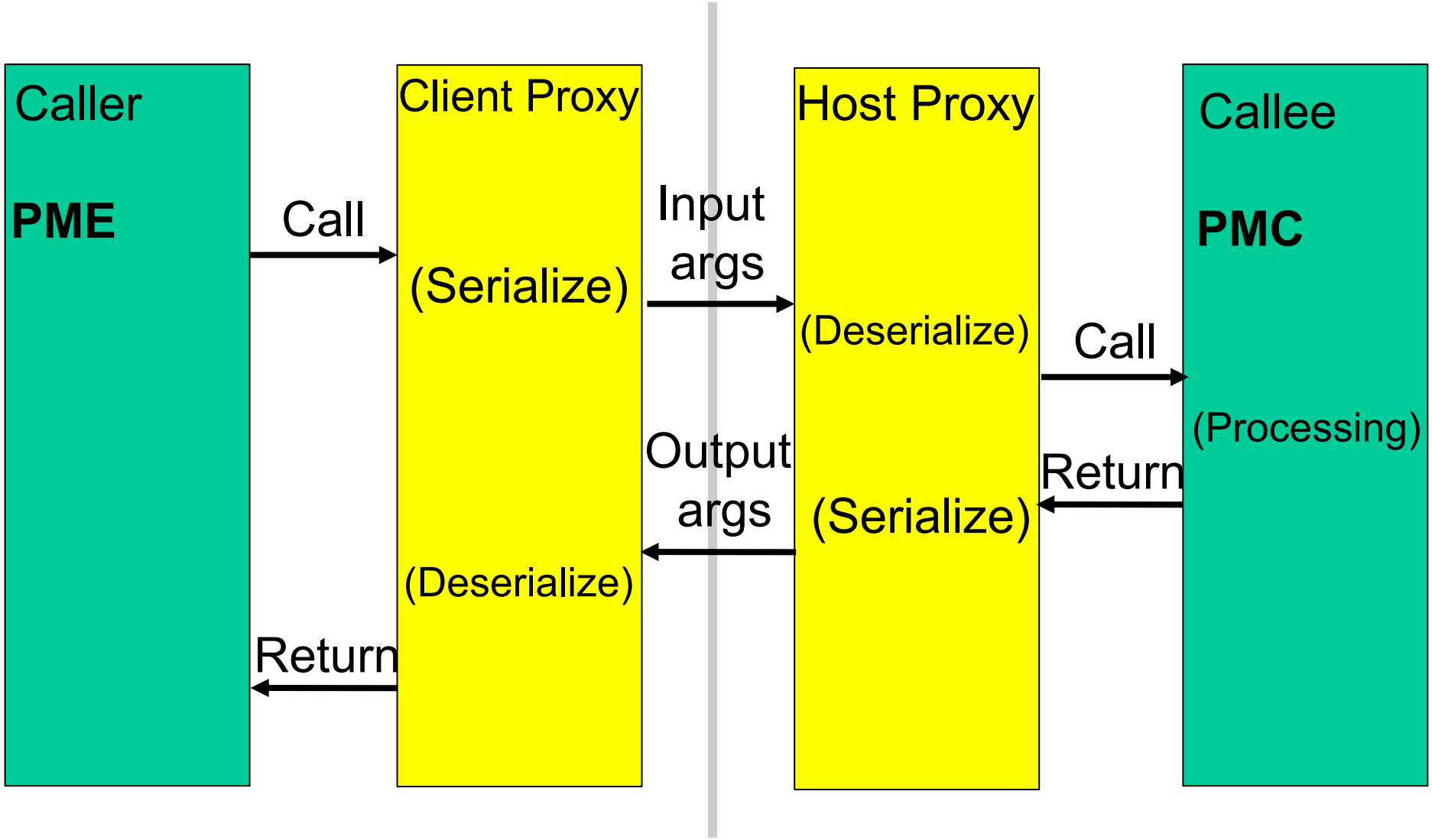
DIRECT CALL



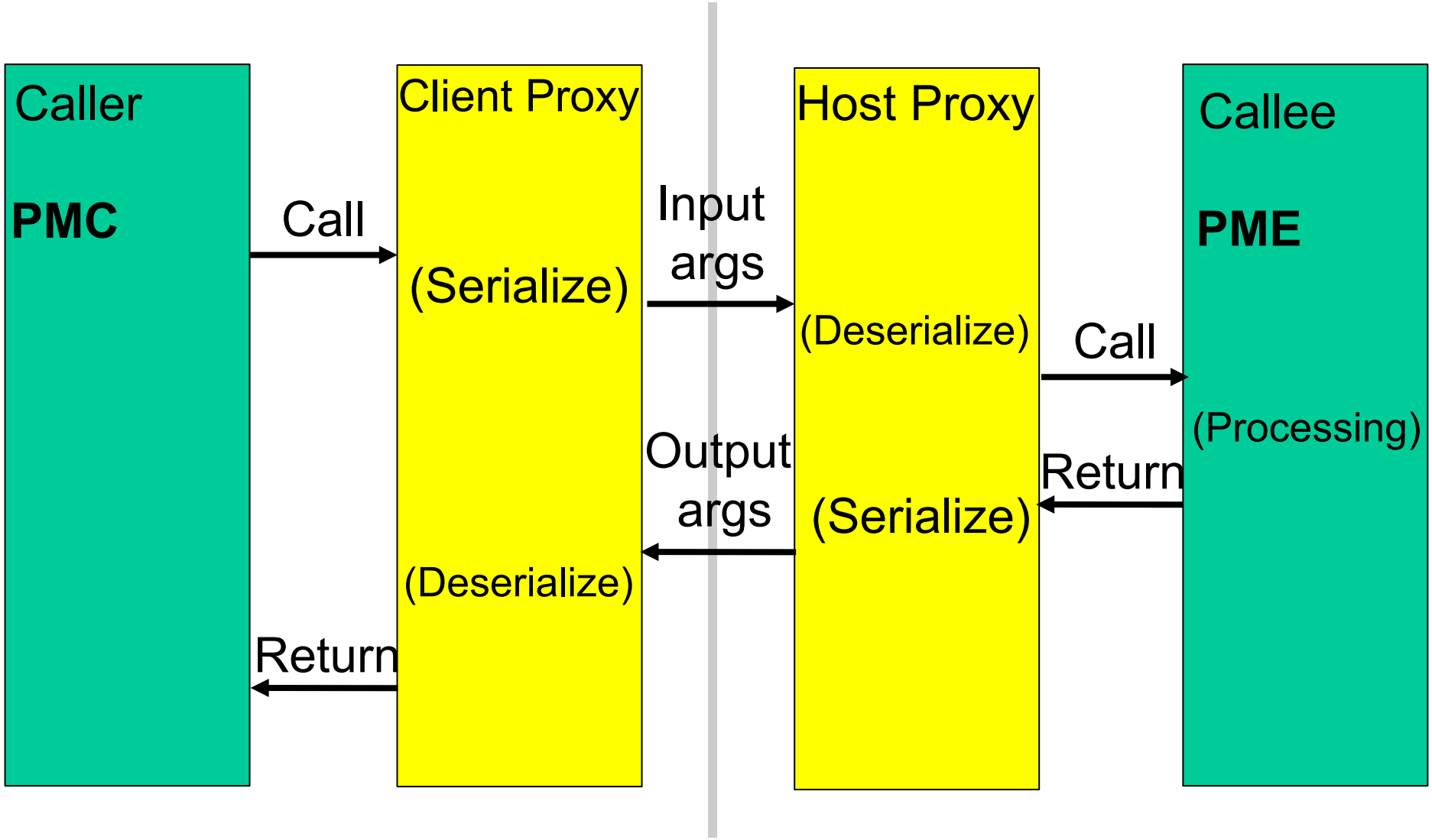
MARSHALING A CALL



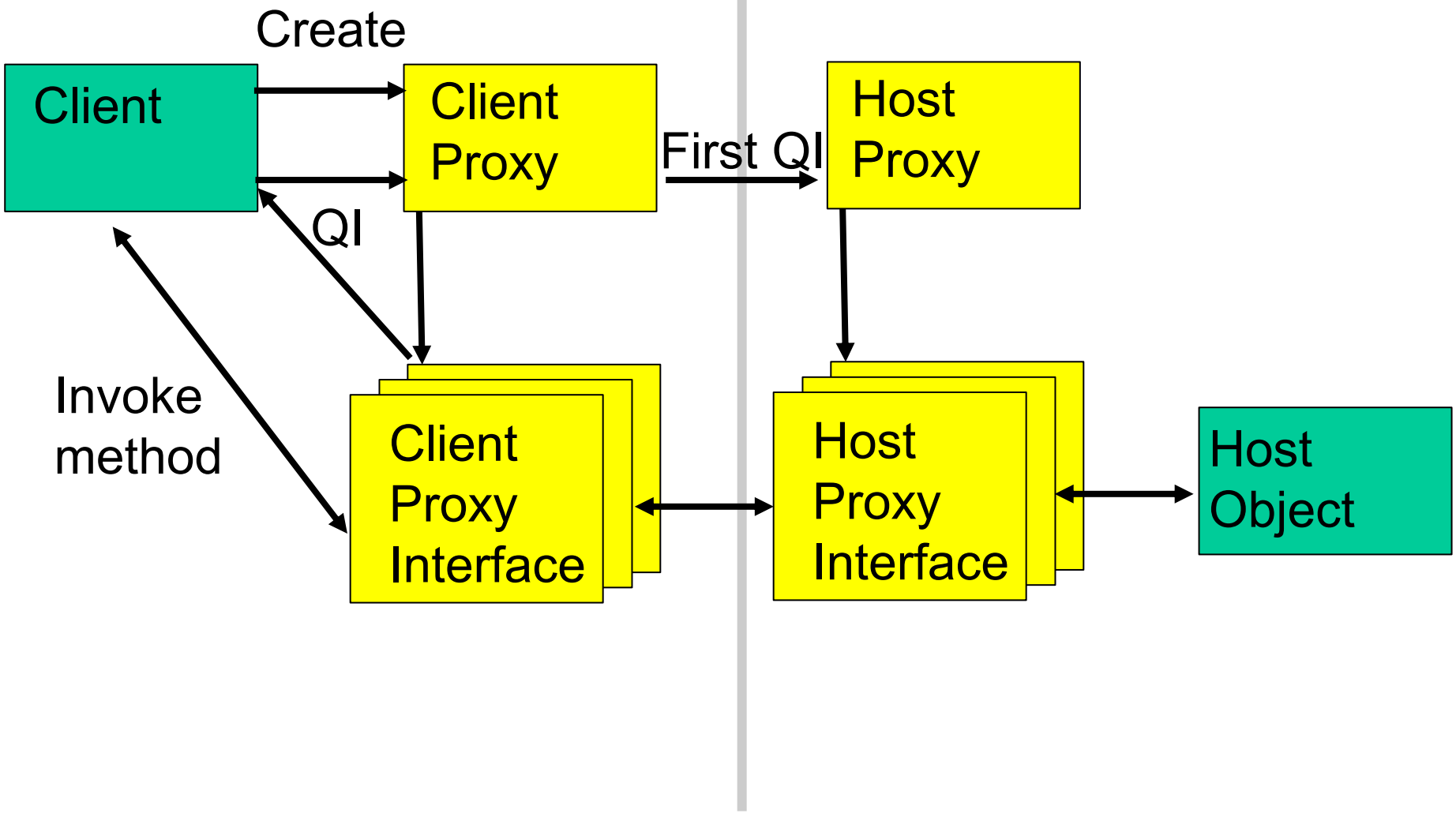
MARSHALING A CALL



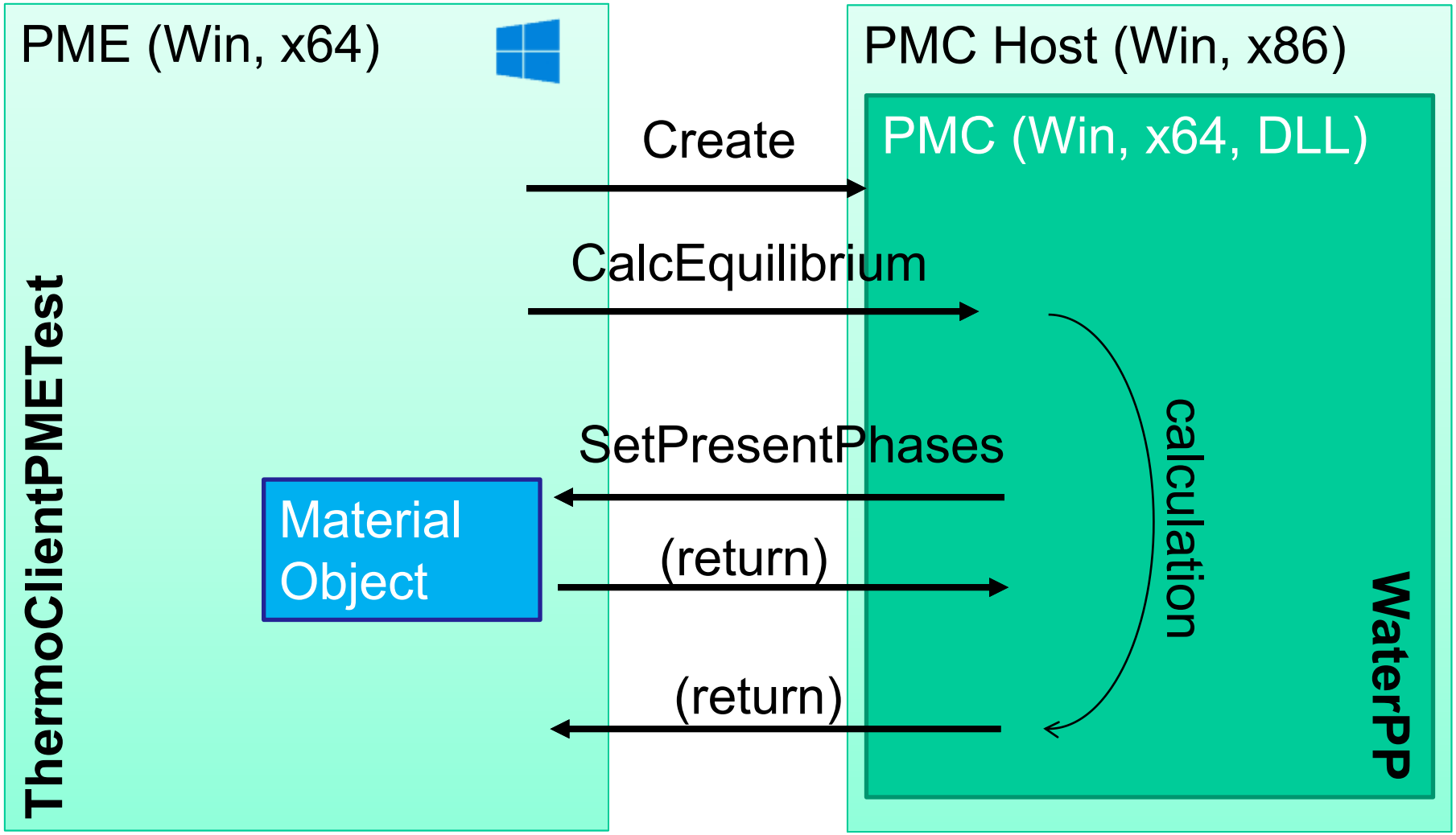
MARSHALING A CALL



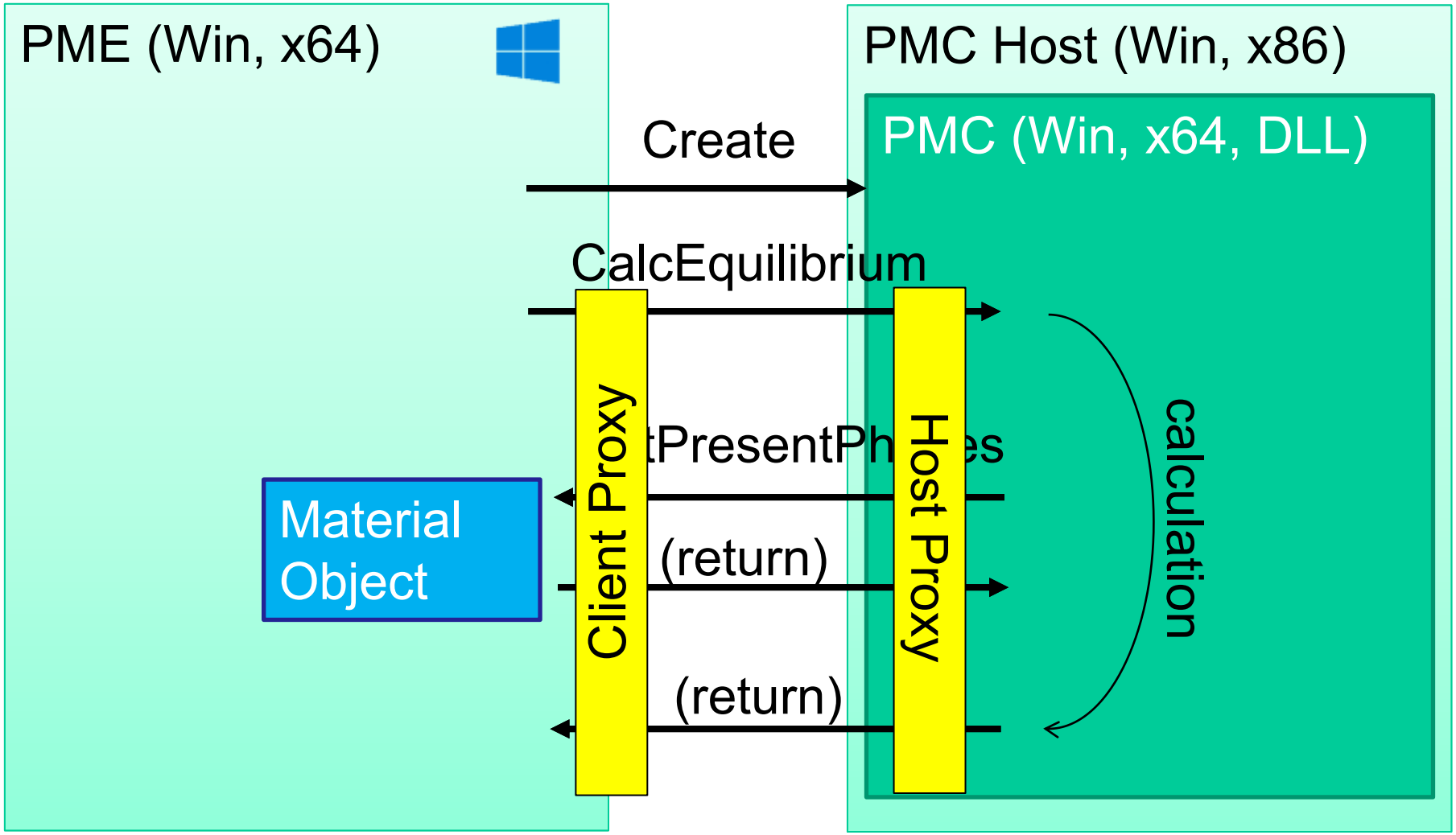
MARSHALING A CALL



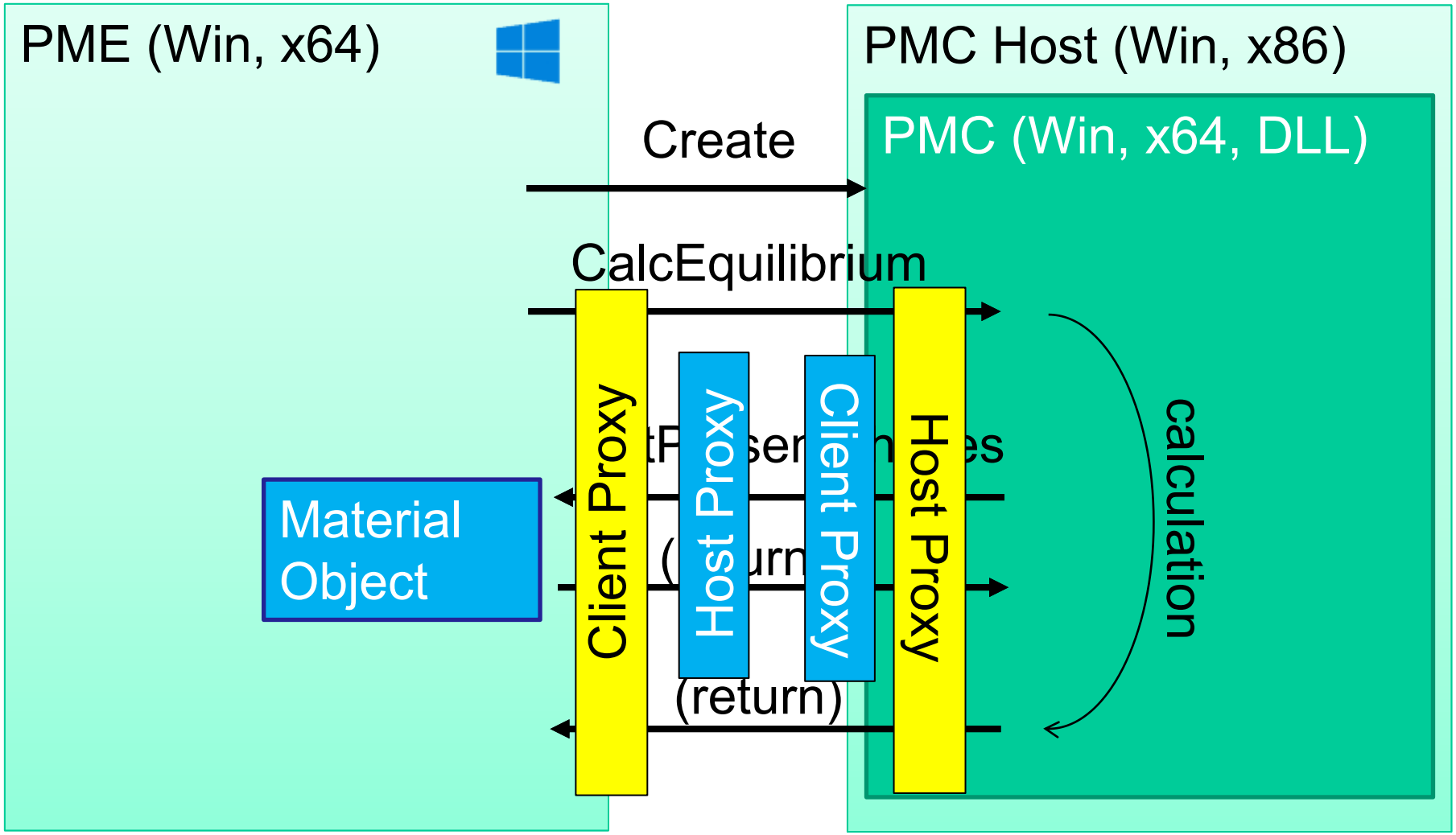
MARSHALING DEMO



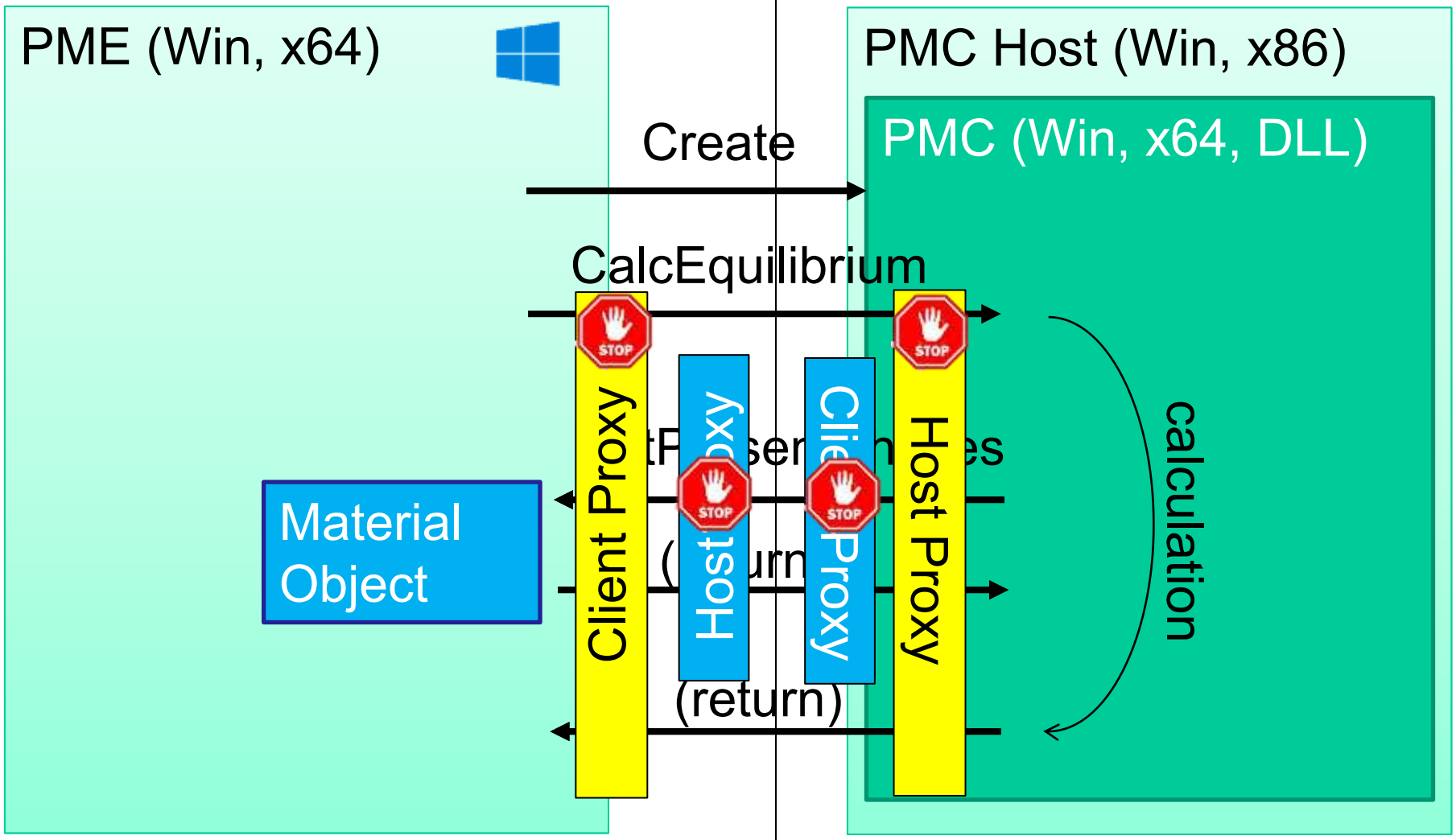
MARSHALING DEMO



MARSHALING DEMO



MARSHALING DEMO



File Edit View Project Build Debug Team Test Analyze Tools Extensions Window Help Search (Ctrl+Q) COBIA

Debug x64 Local Windows Debugger Auto

Solution Explorer

Search Solution Explo

Solution 'COBIA' (1

- ALL_BUILD
- Client
- COBIA
- COBIA_CodeGe
- COBIAIDLParse
- **COBIAMarshal**
- COBIAPMCHos
- cobiaRegister
- COBIAResource
- CodeGeneratio
- CodeGenerator
- CodeGenerator
- COMBIA
- IDL
- IDLInterfaces
  - References
  - External Dep
  - Header Files
    - IDLInterf
    - IDLInterf
    - IDLTypeF
  - CMakeLists.
- ProxyInterfaces
  - References
  - External Dep
  - Header Files
    - ProxyInt
  - Source Files
    - ProxyInt
    - ProxyInt

ProxyInterfaces_CAPEOPEN_1_2.h

Go

ProxyInterfaces_CAPEOPEN_1_2 | Host_Proxy_CAPEOPEN_1_2_ICapeFlowsheetI | ExecuteMethod(int methodIndex, void ** inp

```
5108 class Client_Proxy_CAPEOPEN_1_2_ICapeThermoEquilibriumRoutine :
5109     public Marshal::CapeOpenClientProxyInterfaceBase<Client_Proxy_CAPEOPEN_1_2_ICapeThermoEquilibriumRouti
5110     public ICapeThermoEquilibriumRoutine {
5111
5112     using BaseClass=Marshal::CapeOpenClientProxyInterfaceBase<Client_Proxy_CAPEOPEN_1_2_ICapeThermoEquilib
5113     const CapeUUID &iid;
5114
5115 public:
5116
5117     ~Client_Proxy_CAPEOPEN_1_2_ICapeThermoEquilibriumRoutine() = default;
5118
5119     //ICapeThermoEquilibriumRoutine
5120     static CapeResult COBIAMETHOD _CalcEquilibrium(void *me,/*in*/ ICapeArrayString *specification1,/*in*/
5121         const int methodIndex=4;
5122         Client_Proxy_CAPEOPEN_1_2_ICapeThermoEquilibriumRoutine *This=static_cast<Client_Proxy_CAPEOPEN_1_
5123         This->clearLastError();
5124         CapeResult resultCode;
5125         try {
5126             const Marshal::COBIA_DATA_TYPE inputArgTypes[3]={Marshal::COBIA_DATA_TYPE::CapeArrayStringType
5127             void* inputArgs[3]={specification1,specification2,solutionType};
5128             This->marshaler.ExecuteMethod(This->iid,methodIndex,This,3,inputArgTypes,inputArgs,0,nullptr,n
5129             resultCode=COBIAERR_NoError;
5130         } catch (cape_open_error &ex) {
5131             This->setErrorFromCapeOpenError(COBIATEXT("ICapeThermoEquilibriumRoutine::CalcEquilibrium"),ex
5132             res
5133         } catch
5134             COB
5135             cape_open_error ex1(ex);
5136             This->setErrorFromCapeOpenError(COBIATEXT("ICapeThermoEquilibriumRoutine::CalcEquilibrium"),ex
```

(local variable) Client_Proxy_CAPEOPEN_1_2_ICapeThermoEquilibriumRoutine *This

Search Online

100 % No issues found Ln: 5131 Ch: 83 Col: 92 TABS CRLF

Output

Show output from: Debug

```
The program '[13960] COBIAPMCHost.exe' has exited with code 0 (0x0).
```

Output | Find Symbol Results | Compiler Inline Report | Compiler Optimization Report

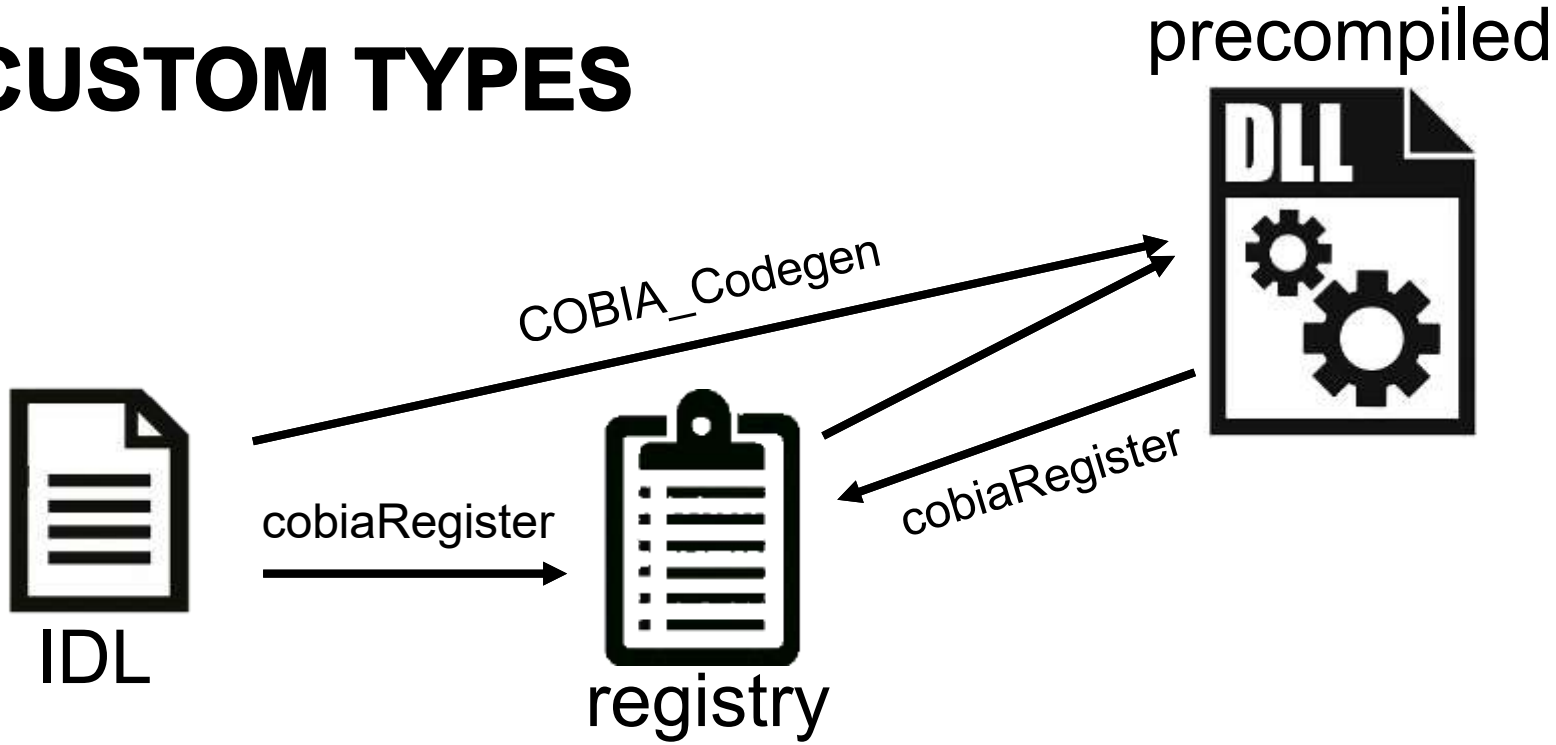
Test Explorer | VA Code Inspection Results | Toolbox | Diagnostic Tools | Server Explorer | Notifications | Properties

Ready Add to Source Control

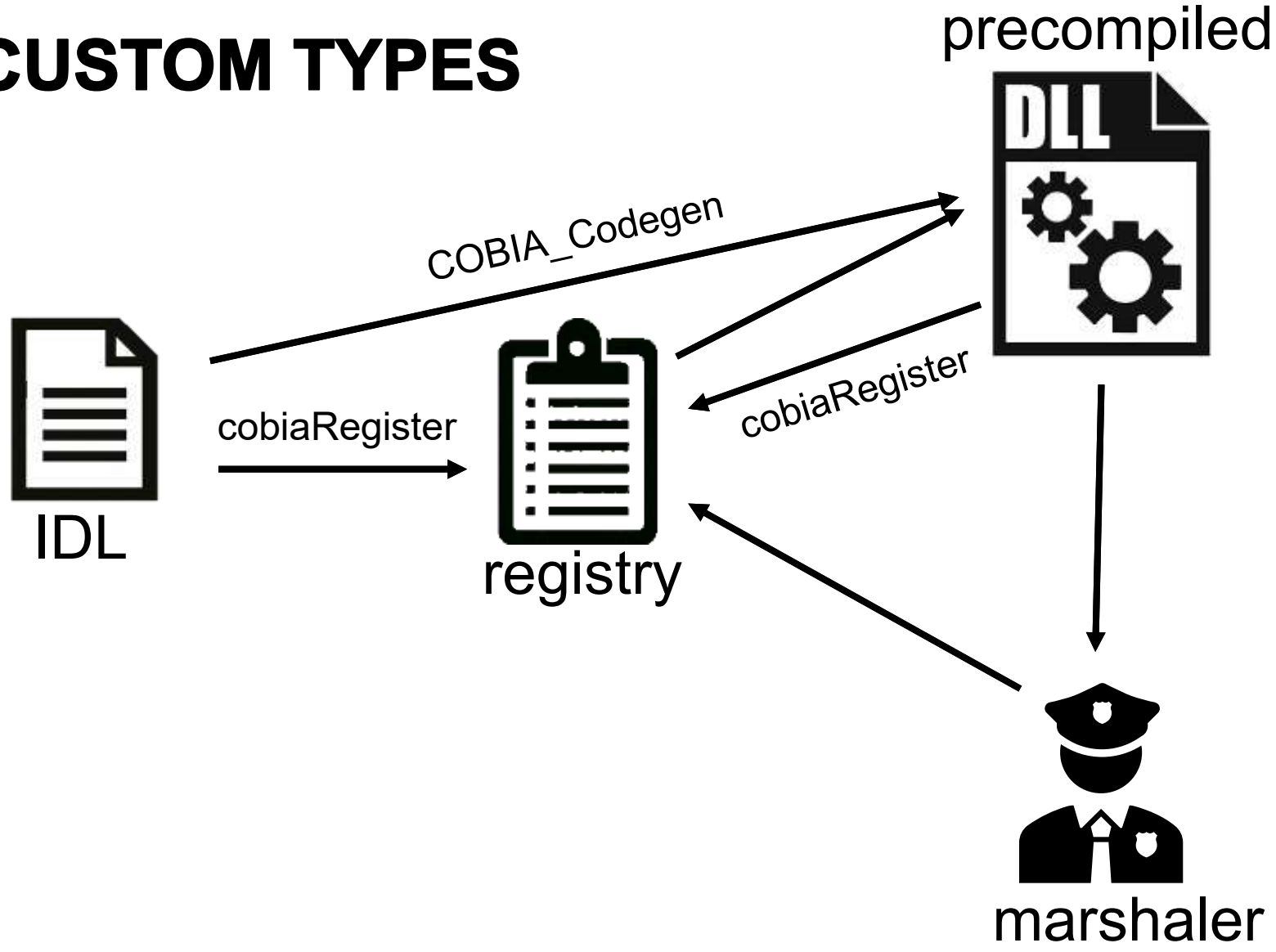
PROXY OBJECTS

- Precompiled by COBIA
- Type information in registry
  - Many benefits in code generation!
- Precompiled by software vendor
  - From type information, code generators
- Compiled on the fly
  - Native: libffi (prototype available)
  - .NET, java: reflection (.NET prototyped)

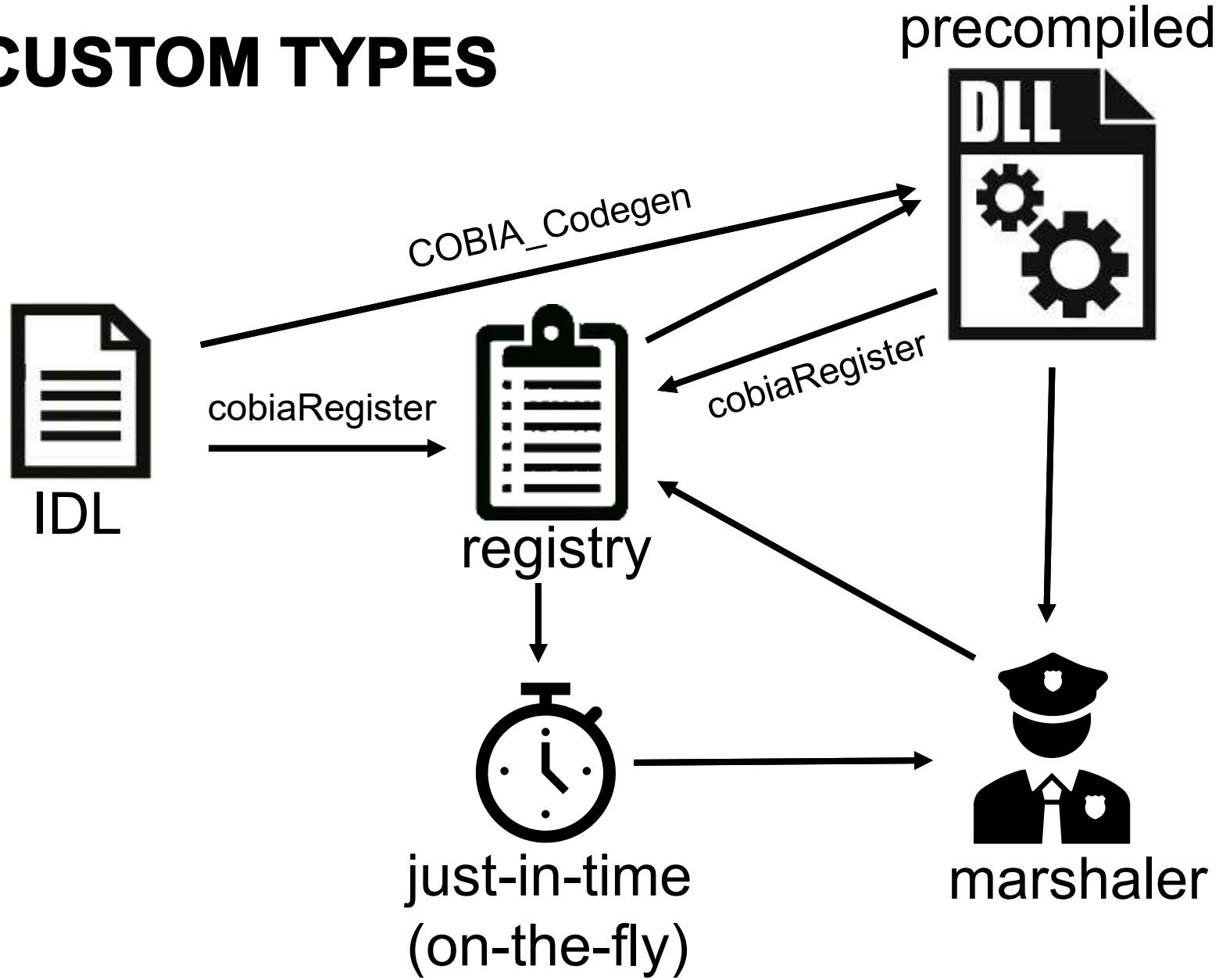
CUSTOM TYPES



CUSTOM TYPES



CUSTOM TYPES



THE COBIA THREADING MODELS

- *unrestricted threading:* the PME can access PMC object from any thread, as long as it takes care not to do so concurrently
- *restricted threading:* a PMC can be accessed only from the thread in which it was created.

THE COBIA THREADING MODELS

- *unrestricted threading:* the PME can access PMC object from any thread, as long as it takes care not to do so concurrently
- *restricted threading:* a PMC can be accessed only from the thread in which it was created.

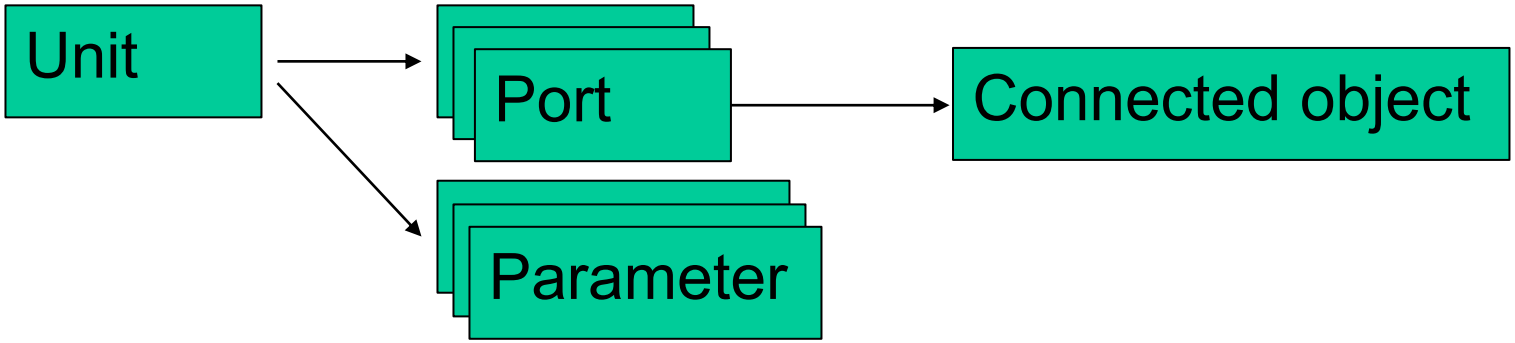
Restricted threading is not single threading!

THE COBIA THREADING MODELS

- *unrestricted threading:* the PME can access PMC objects from any thread, as long as it takes care **not** to do so **concurrently**
- *restricted threading:* a PMC can be accessed only from the thread in which it was created.

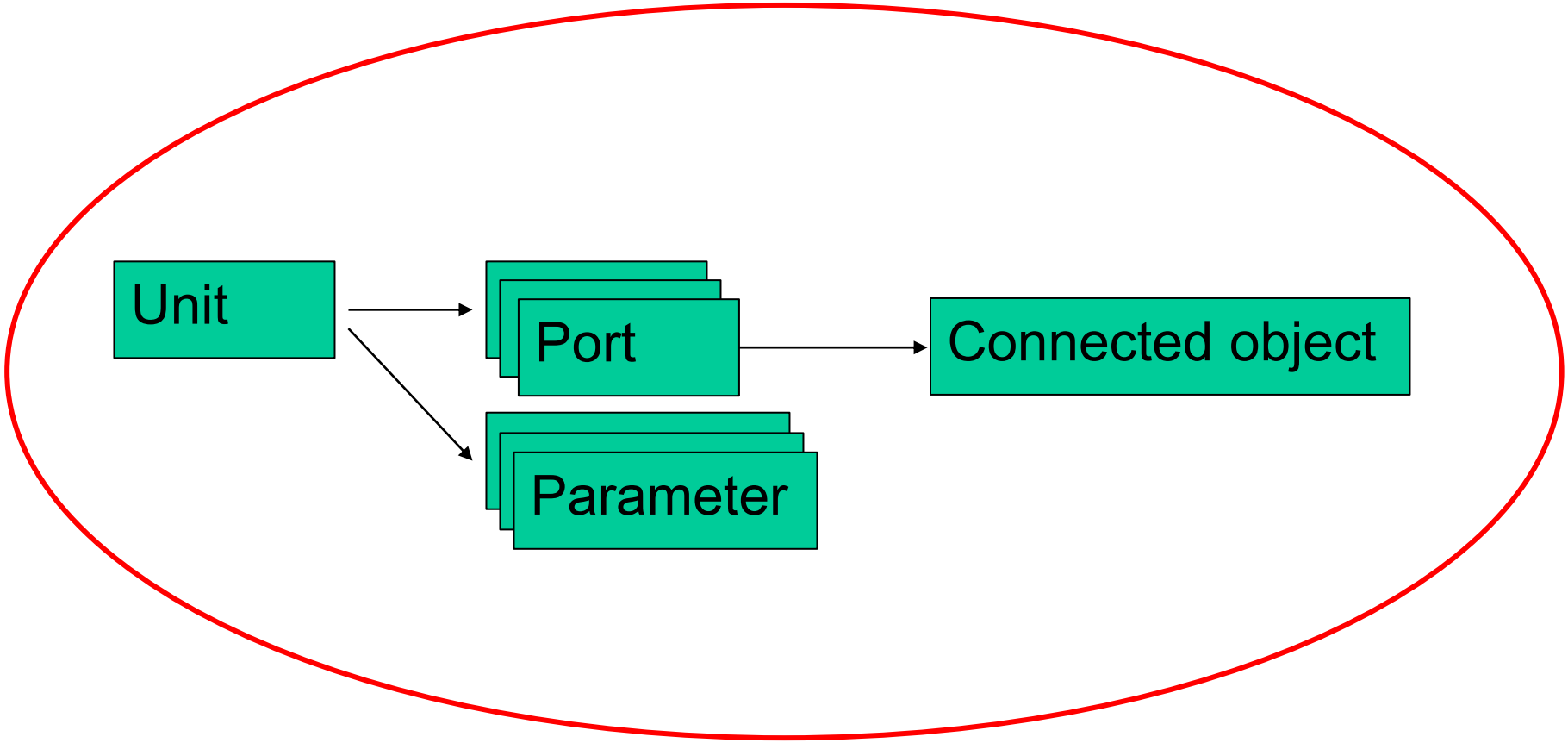
THE COBIA THREADING MODELS

- *unrestricted threading:* **not concurrently**



THE COBIA THREADING MODELS

- *unrestricted threading:* **not concurrently**



THE COBIA THREADING MODELS

- *unrestricted threading:* the PME can access PMC objects from any thread, as long as it takes care **not** to do so **concurrently**
- *restricted threading:* a PMC can be accessed only from the thread in which it was created.
  - PME can indicate that it does *not need* unrestricted threading!
  - if the PME does not do so, COBIA will marshal the PMC in-process

COBIA THREADING VS COM THREADING

- COBIA PMCs are fully compliant with any COM threading model. No problems here.
  - in a COM Single Threaded Apartment (STA), COBIA will create the PMC without the need for unrestricted threading
  - in a COM Multi-Threaded Apartment (MTA), COBIA will create the PMC with marshaling if needed.

COBIA THREADING VS COM THREADING

- COM PMCs are not necessarily compliant with COBIA threading
  - when a COM object is created, COM needs to be initialized for the current thread with a COM threading model
  - the problem is that COM threading models are per thread, whereas COBIA threading can be indicated per PMC.

STATUS

PME (Win, x64)

PMC (Win, x64, DLL)

Thread

PMC (Win, x64, DLL)

Native - .NET

PMC (C#, DLL)

Logging:

PMC (Win, x64, DLL)

PMC Host (Win, x86)

PMC (Win, x86, DLL)

PMC (Win, x64, EXE)

TCP/IP

SSH

Somewhere else:

PMC Host (Linux)

PMC (Linux, so)

LINUX – WINDOWS INTEROP

Command Prompt

```
C:\Users\jasper>\werk\co\PhaseIII\COBIA\x64\bin\debug\cobiaRe
gister -i {cb748fe1-4cfa-4bd0-bc42-4cc1b63a50de}
```

jasper@Jmain: ~

```
jasper@Jmain:~$
```

CONCLUSIONS

- Marshaling is a very important step in COBIA phase III development, and a proof-of-concept is delivered
- Some details on COM-COBIA threading model interoperability still remain to be worked out.
- COBIA was already pretty cool, and just became a lot cooler!

REMINDER: PLEASE TRY!

- Source is available to CO-LaN members
  - /trunk: Phase II, COBIA 1.2.0.8
  - /branches/phaseIII: Phase III work-in-progress
- COBIA wizard available
- Trying is encouraged!
- Feedback is welcome…