

Full Deployment: COBIA is production ready

Jasper van Baten - AmsterCHEM

Bill Barrett – EPA

Michael Hlavinka- Bryan Research & Engineering

Kyle Abrahams

8 October 2024

Why COBIA

- ◆ **CAPE-OPEN is OPEN**
 - ⇒ COM is only available on Windows
 - ⇒ COBIA: Windows + Linux + MacOS
- ◆ **Strong typed:**
 - ⇒ Easier to program (no type checking)
 - ⇒ Less error prone (incorrect type not possible)
- ◆ **Memory ownership model**
 - ⇒ Easier to program (caller owns all memory)
 - ⇒ Less error prone (no confusion about ownership)
 - ⇒ More efficient (easy to recycle memory)

Overview

- ◆ **Phase III: COBIA development phases**
- ◆ **Threading model reminder**
- ◆ **Author COBIA types, expose to COM**
- ◆ **Marshaling for custom types**
- ◆ **Language bindings**
- ◆ **Platforms**
- ◆ **Transport**
- ◆ **Logging**
- ◆ **Closing remarks**

COBIA development phases

- ◆ Phase I: proof of concept. Core functionality only, in-process, C++ only, Windows only
- ◆ Phase II: full interface set, C++ only, Windows only (but Linux functional)
- ◆ Phase III: Cross platform Interop:
 - ⇒ **Marshaling**
 - ⇒ Language bindings
 - ⇒ Remote
 - ⇒ Logging
- ◆ Phase IV: Documentation

COBIA development phases

◆ Phase III: Cross platform Interop:

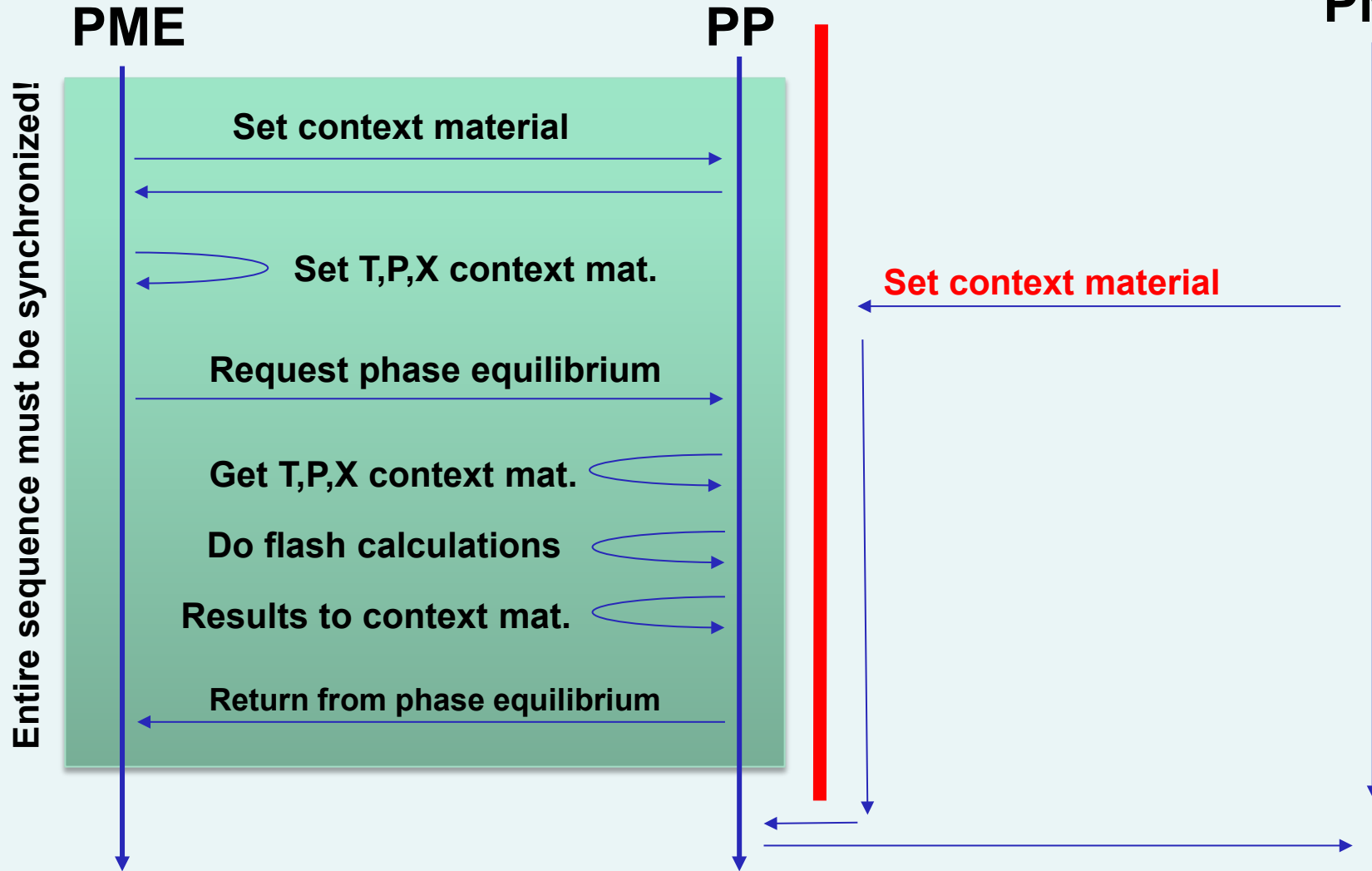
⇒ **Marshaling**

- Threading models
- COM marshaling
- Type consistency
- Custom interfaces
- Proxy generation
 - on-the fly proxies
 - precompiled
- Type registration
- Type API

CAPE-OPEN interoperability is not stateless!

Thread 1

Thread 2
PME

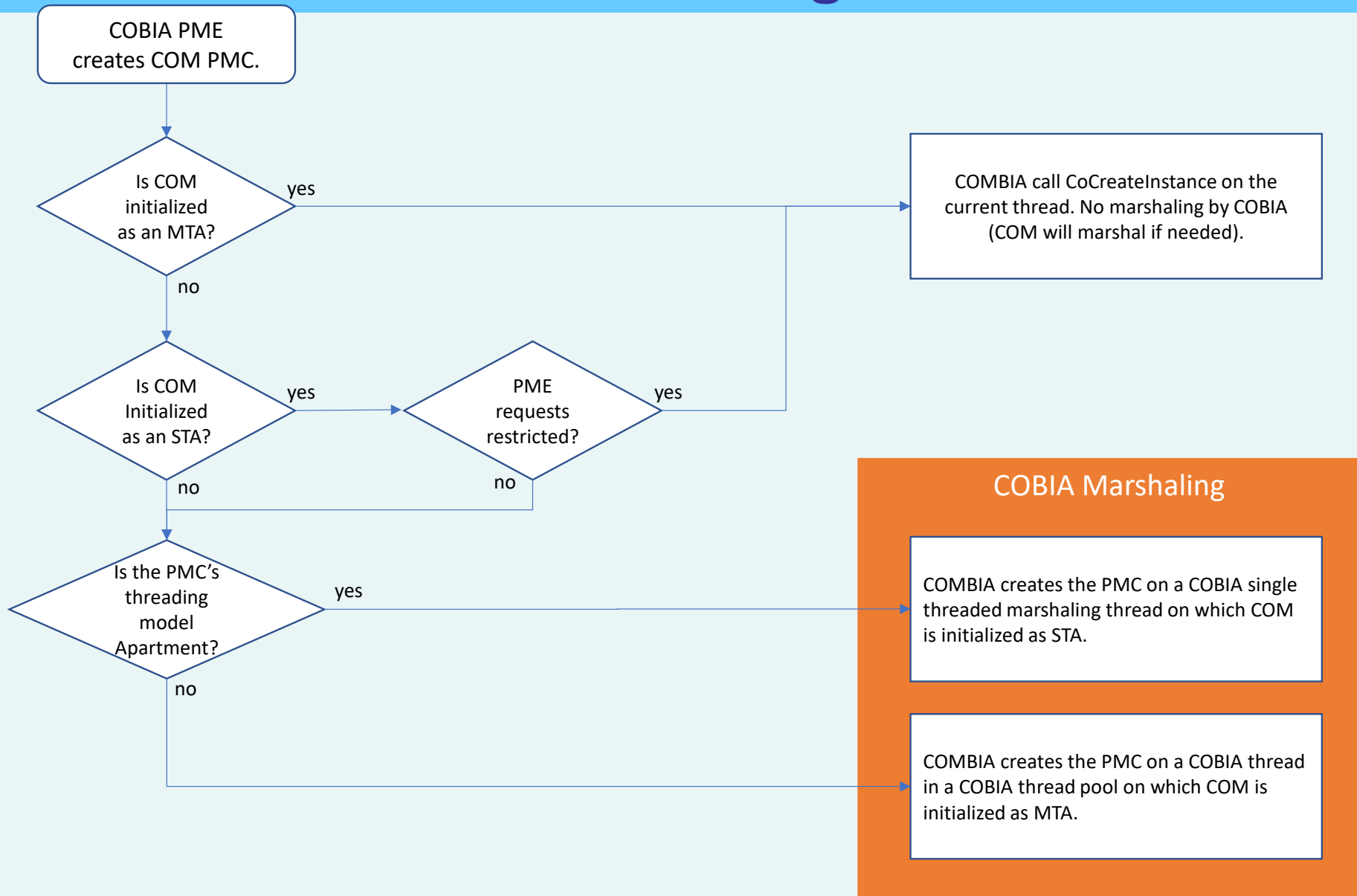


COBIA threading models



- **COBIA threading model per PMC – PME can mix and match different threading models per thread**
- **DEFAULT:**
 - **PME can call PMC from any thread**
 - **PME may not make concurrent calls on PMC**
 - **... or any of its secondary objects!**
- **RESTRICTED:**
 - **PME can only call PMC from one thread**
- **PME states intent on PMC creation: DEFAULT or RESTRICTED**
 - **RESTRICTED PMC in DEFAULT context is marshaled**

Threading



COBIA Marshaling

COMBIA creates the PMC on a COBIA single threaded marshaling thread on which COM is initialized as STA.

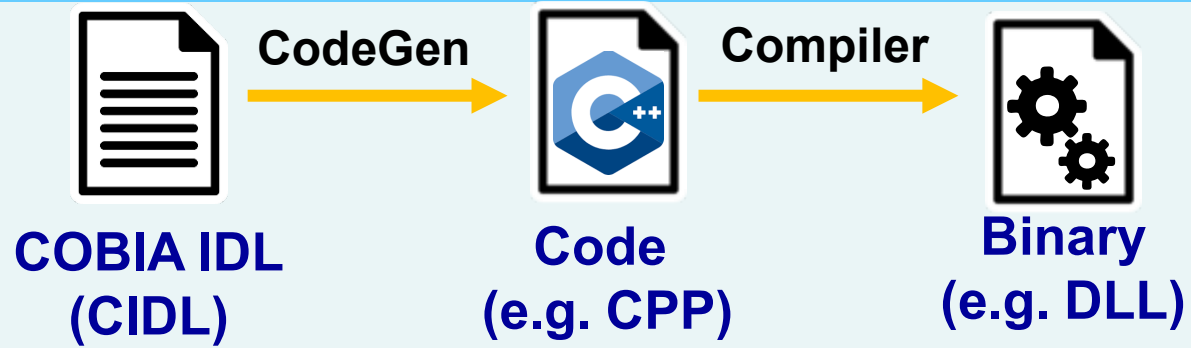
COMBIA creates the PMC on a COBIA thread in a COBIA thread pool on which COM is initialized as MTA.

Developer notes for threading

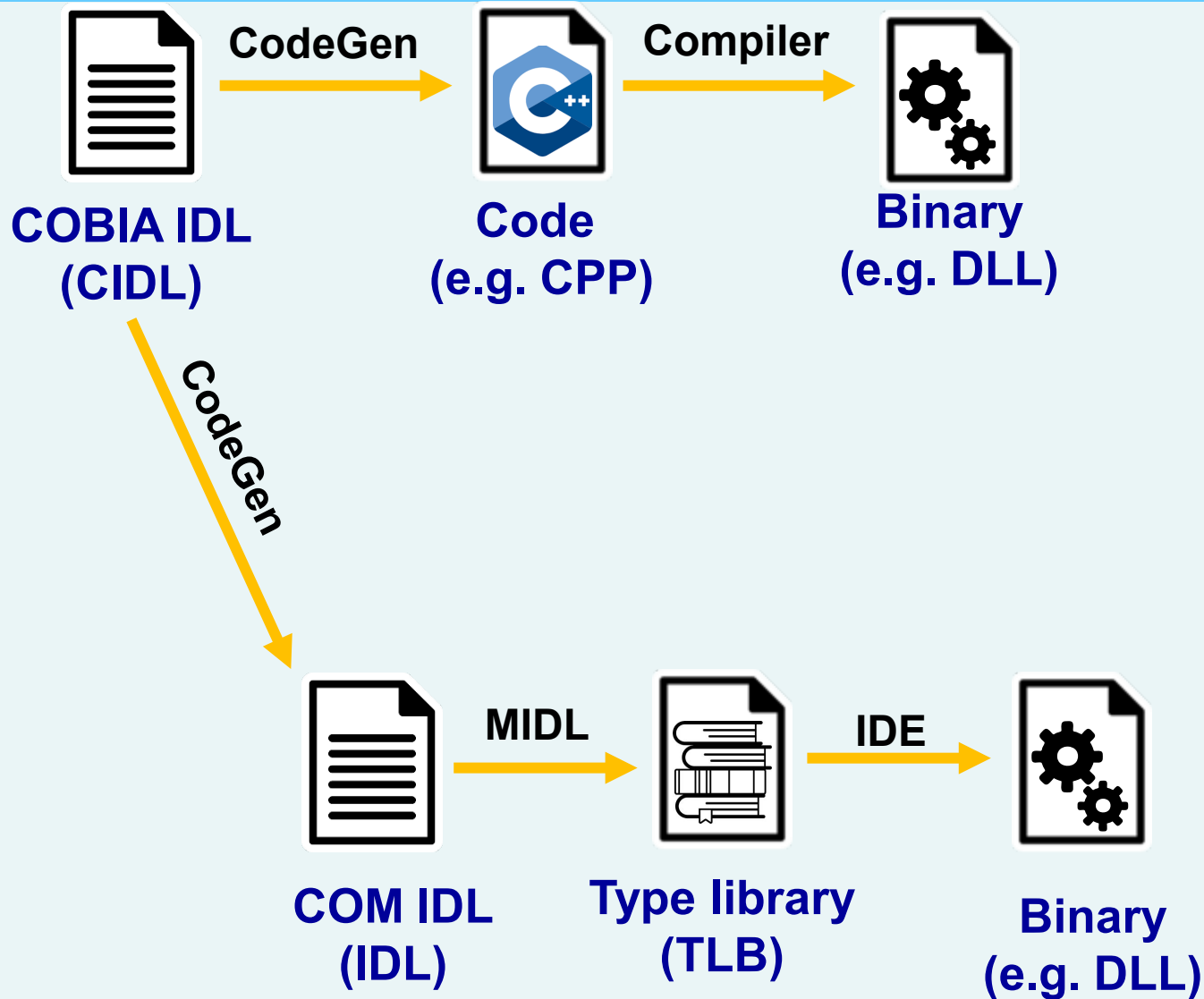
- ❑ **If you are a PMC developer: adhere to COBIA default threading model where possible. No further action needed.**

- ❑ **If you are a PME developer:**
 - ⇒ **Tell COBIA your intent: specify threading model creation flags (if not, COBIA will assume worst case and will marshal if needed)**
 - ⇒ **On Windows: Initialize COM appropriately when possible (if not, COM objects will be marshaled)**

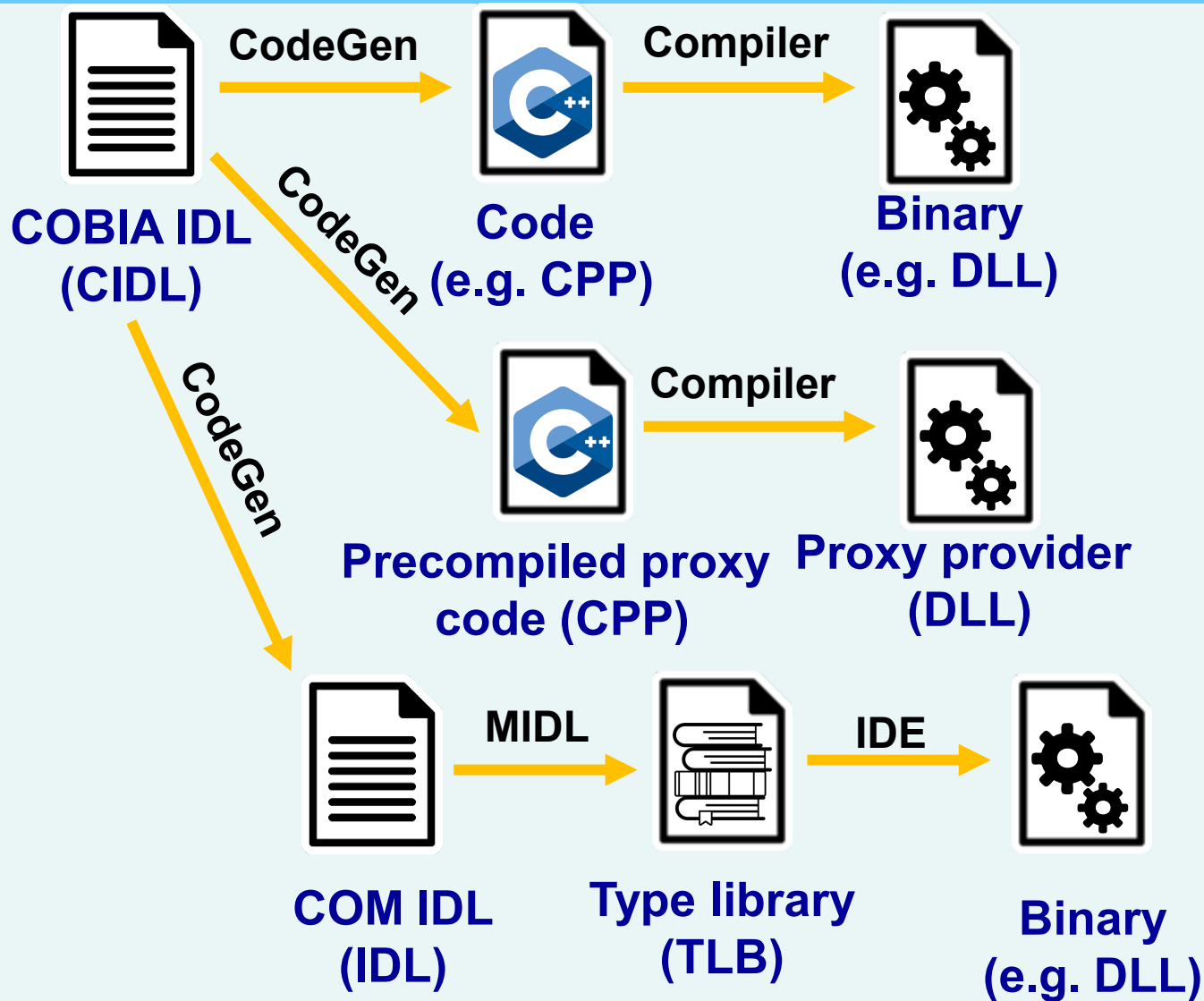
Author COBIA types, expose to COM



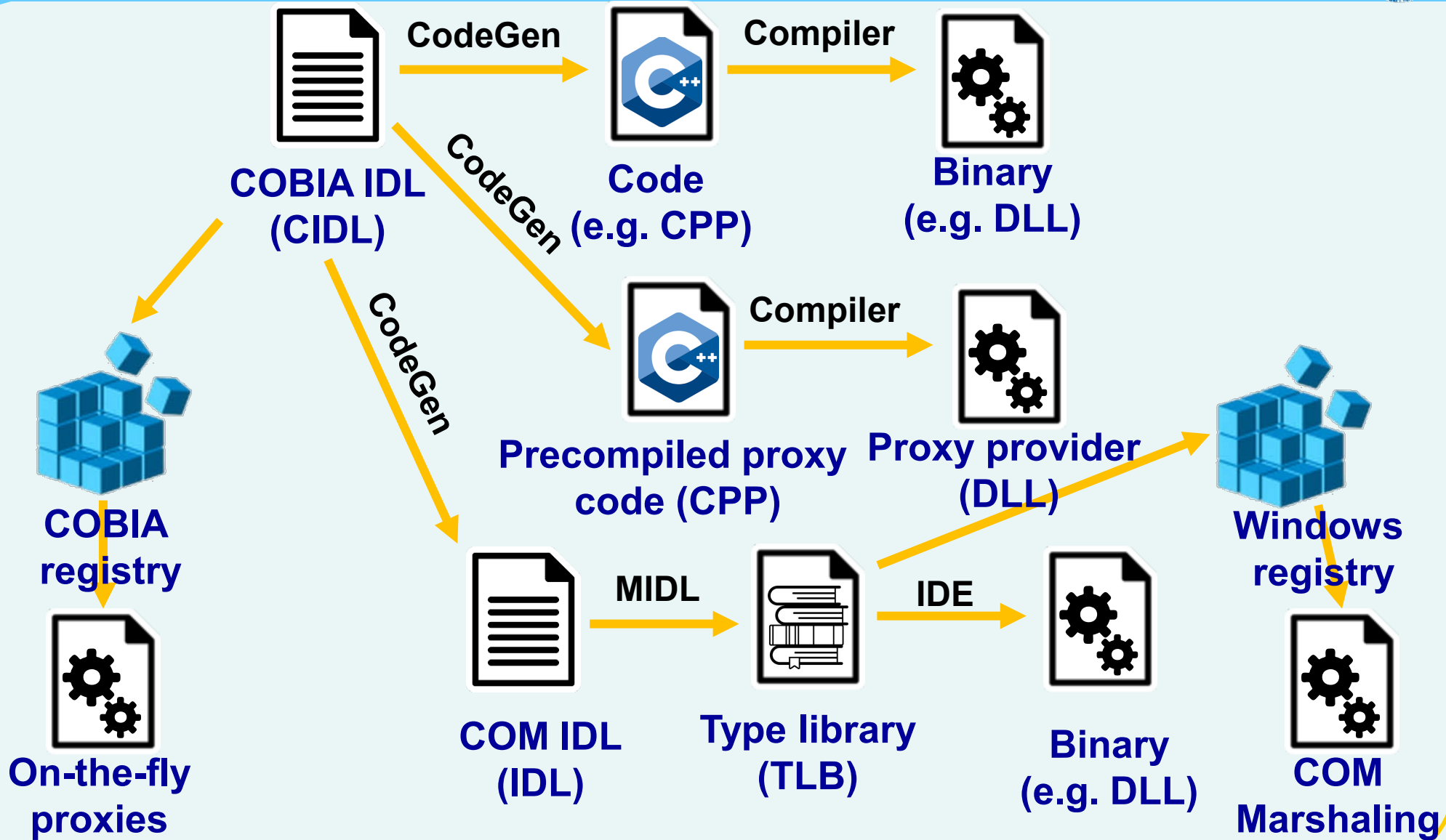
Author COBIA types, expose to COM



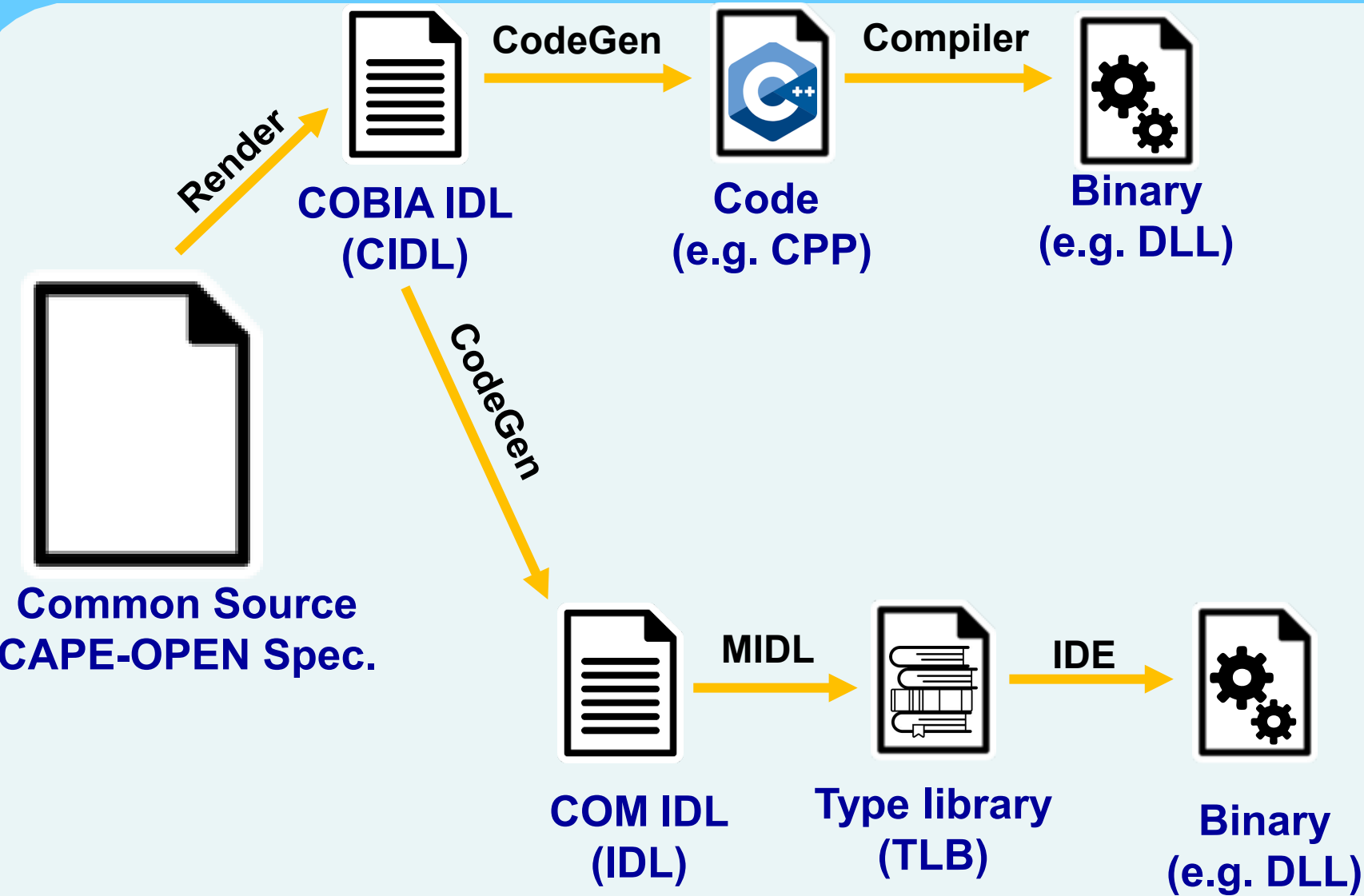
Author COBIA types, expose to COM



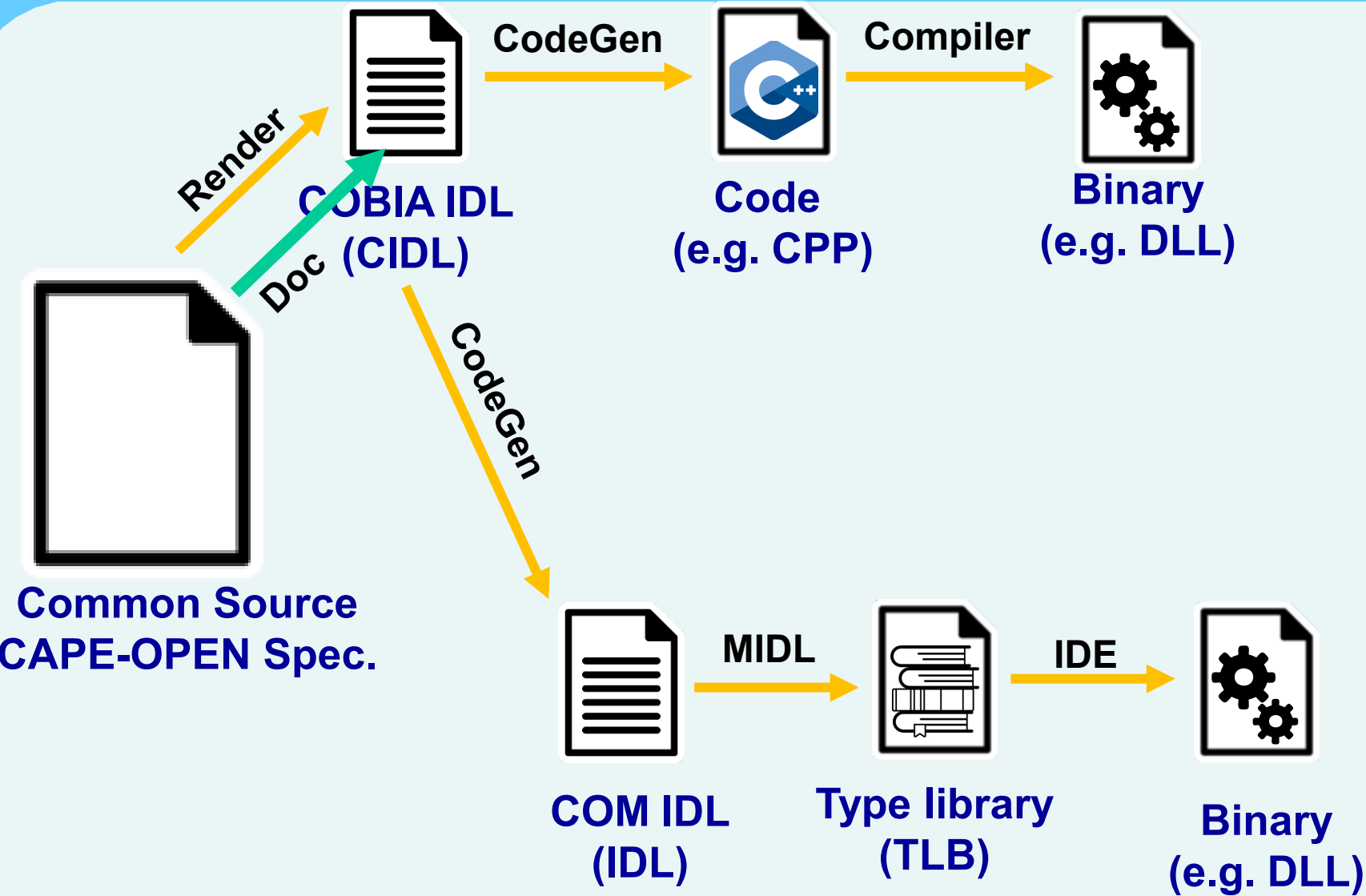
Author COBIA types, expose to COM



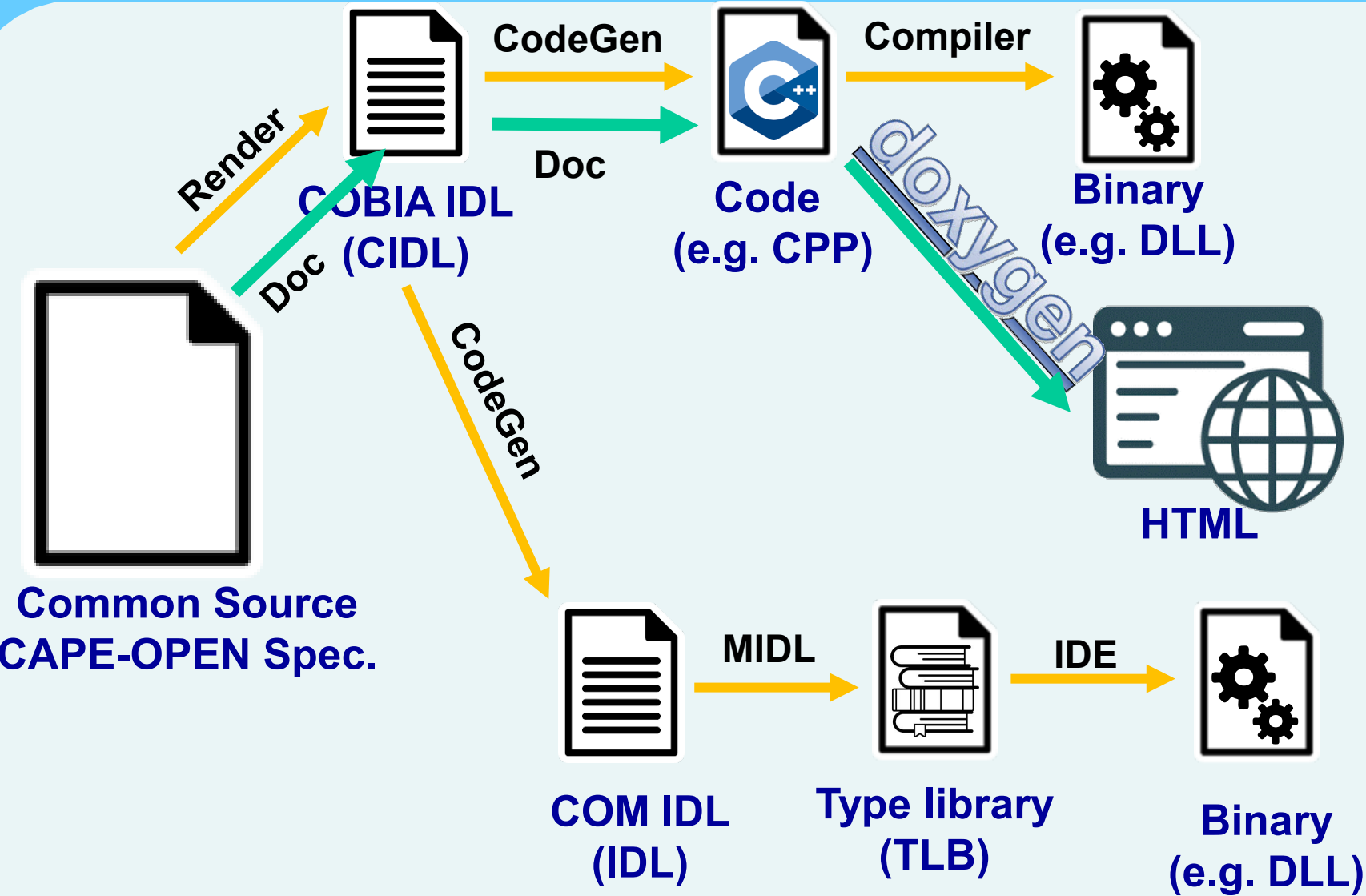
Author COBIA types, expose to COM



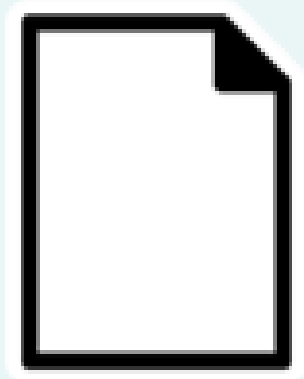
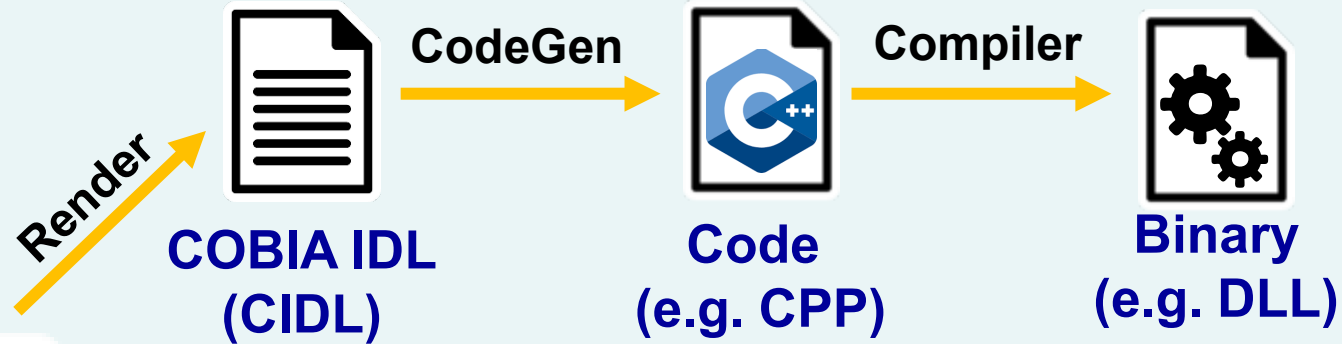
Author COBIA types, expose to COM



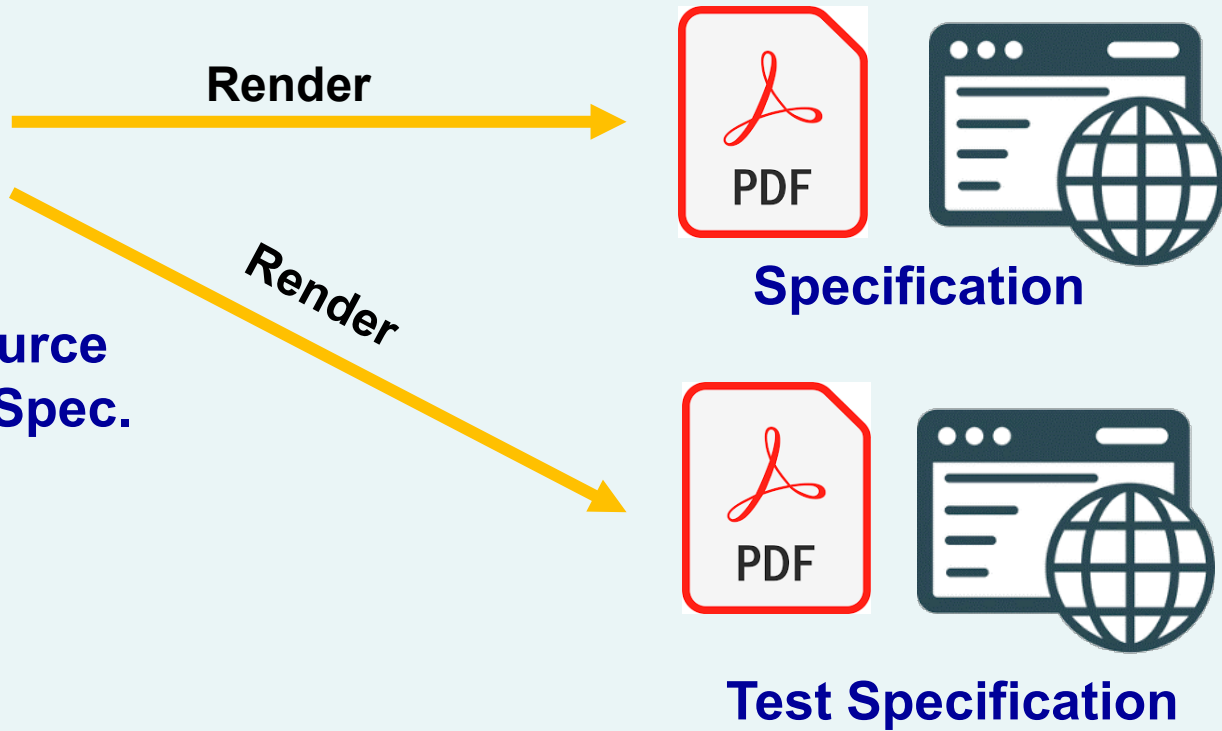
Author COBIA types, expose to COM



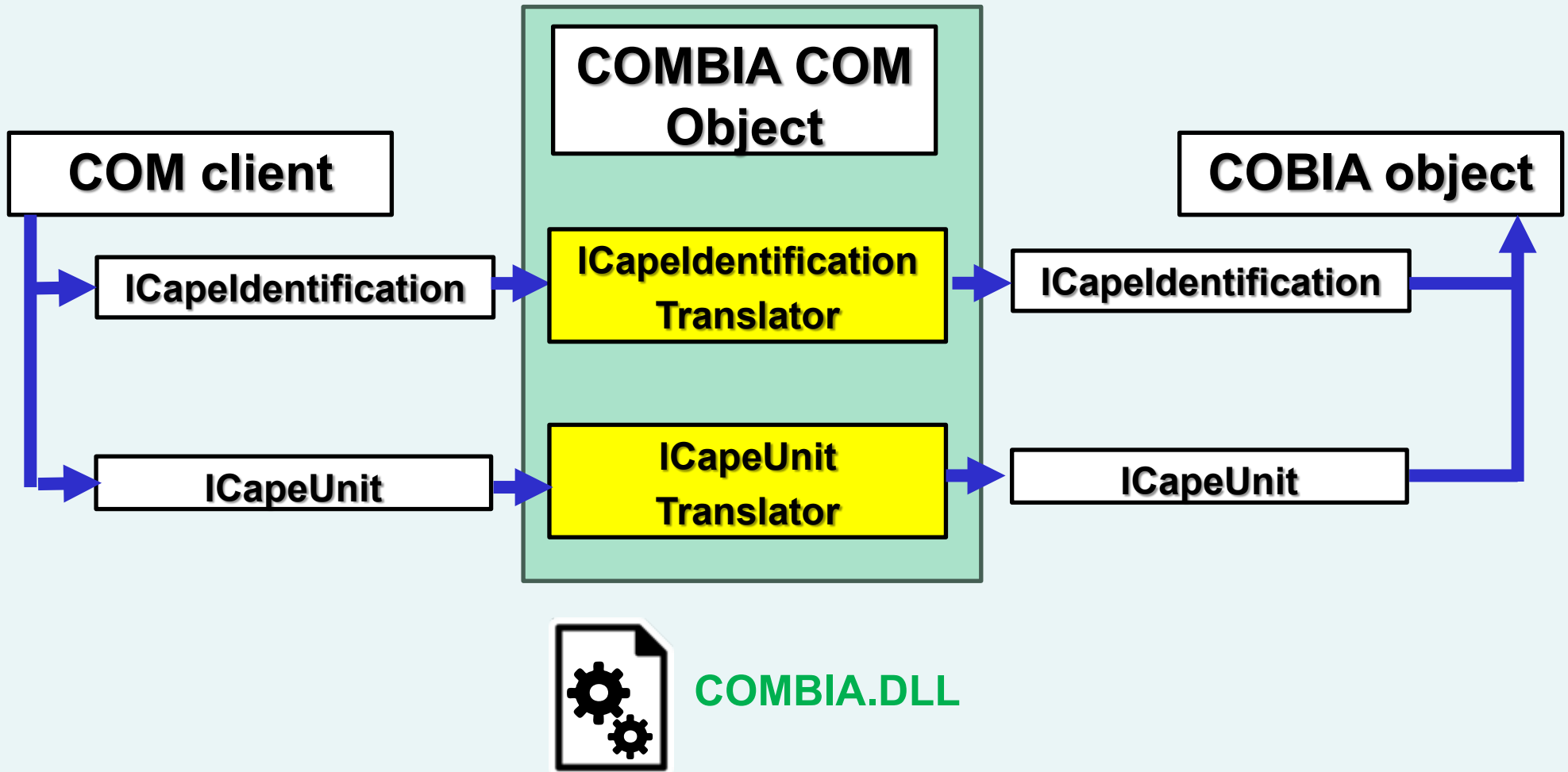
Author COBIA types, expose to COM



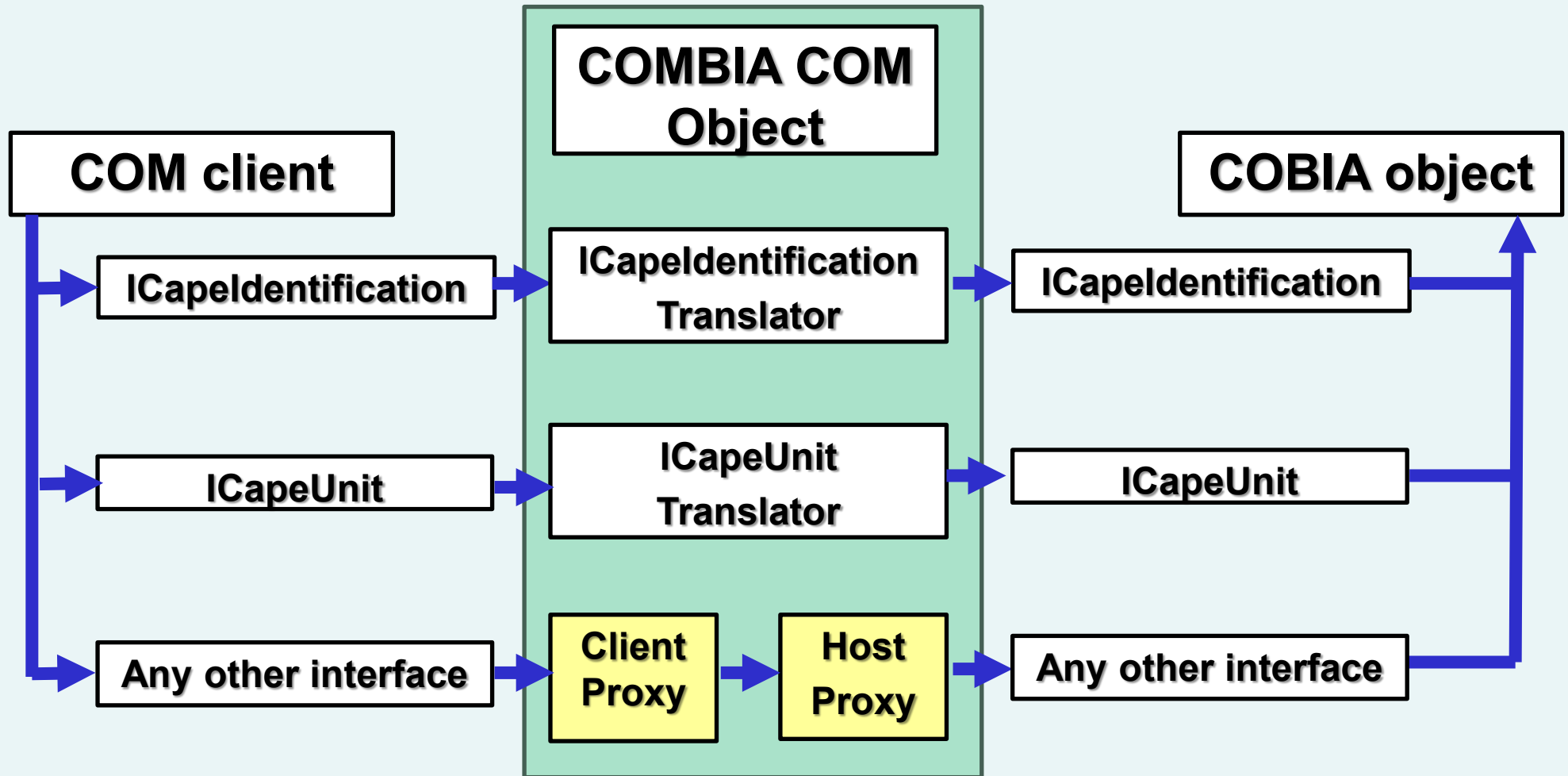
Common Source
CAPE-OPEN Spec.



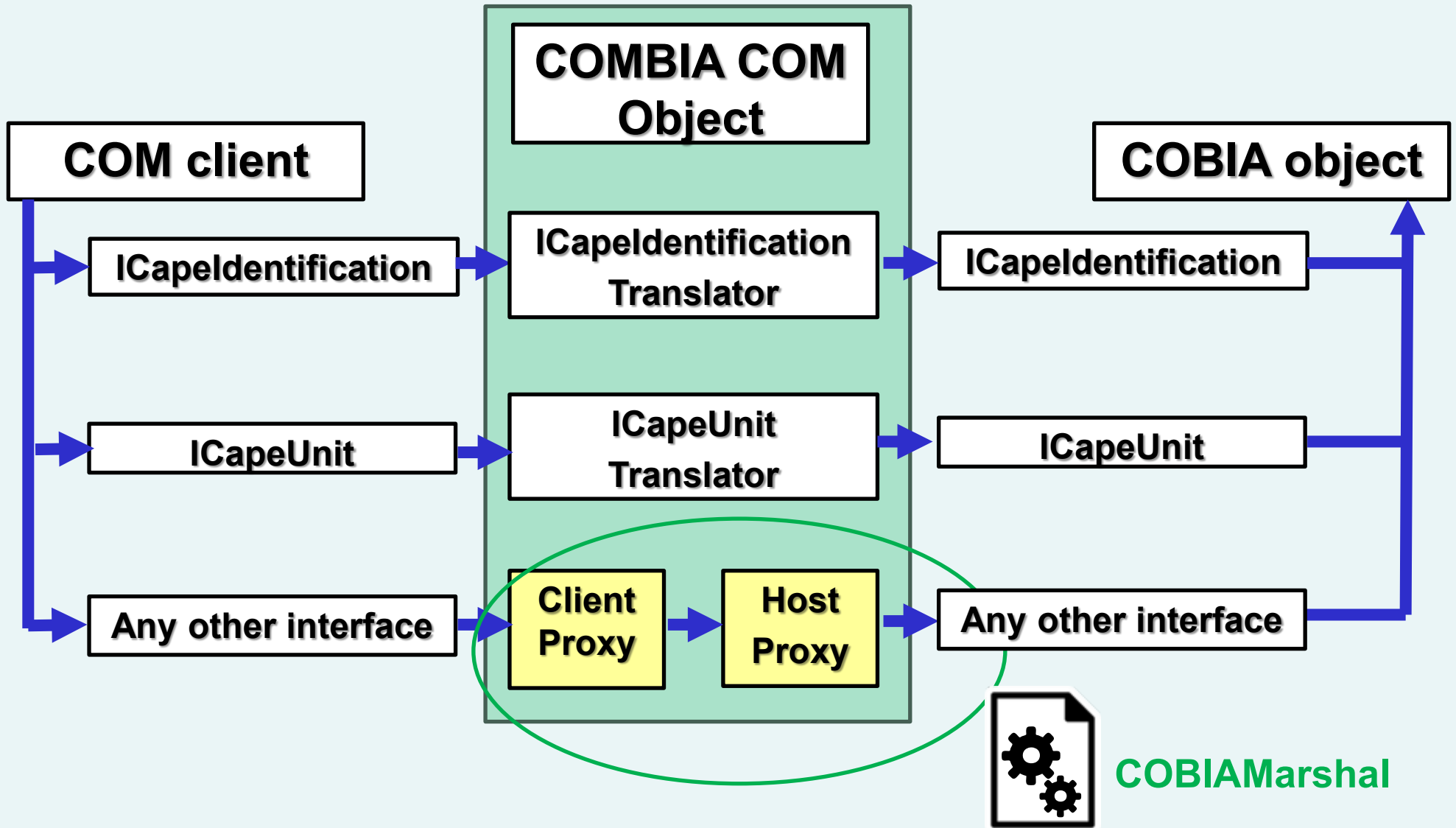
Author COBIA types, expose to COM



Author COBIA types, expose to COM



Author COBIA types, expose to COM





Author COBIA types, expose to COM



- Marshaling based on type info only works if COM and COBIA types are consistent
- Abolished `IDispatch` (see presentation 2014)
- No late binding
- Interfaces derive from `IUnknown`
- Interfaces marked `[OLEAutomation]`
 - Needed for default marshaler compatibility
- Types must be `OLEAutomation` compatible
 - No UUID; marshaled as `BSTR`
- Enums of known size: `[v1_enum]`

COBIA Language bindings

- C++** 
- ISO-C (1999)** 
 - The C-ABI is used for interoperability**
 - Coming up: Rust (AmsterCHEM)**
 - Pathway to other language bindings, such as F90**

- .NET:**
 - .NET vs .NET Framework (Windows only)**
 - Need a business case**
 - Path on calling .NET from native needs further investigation**

- Other language bindings: need business case**
 - Pyhon, java, ...**

COBIA's Platforms



Windows (since Phase I)



Linux (since Phase II)



- Revision needed for default folders
- WindowID to be sorted
- Planning to distribute as .deb, .rpm (2024/2025)



MacOS



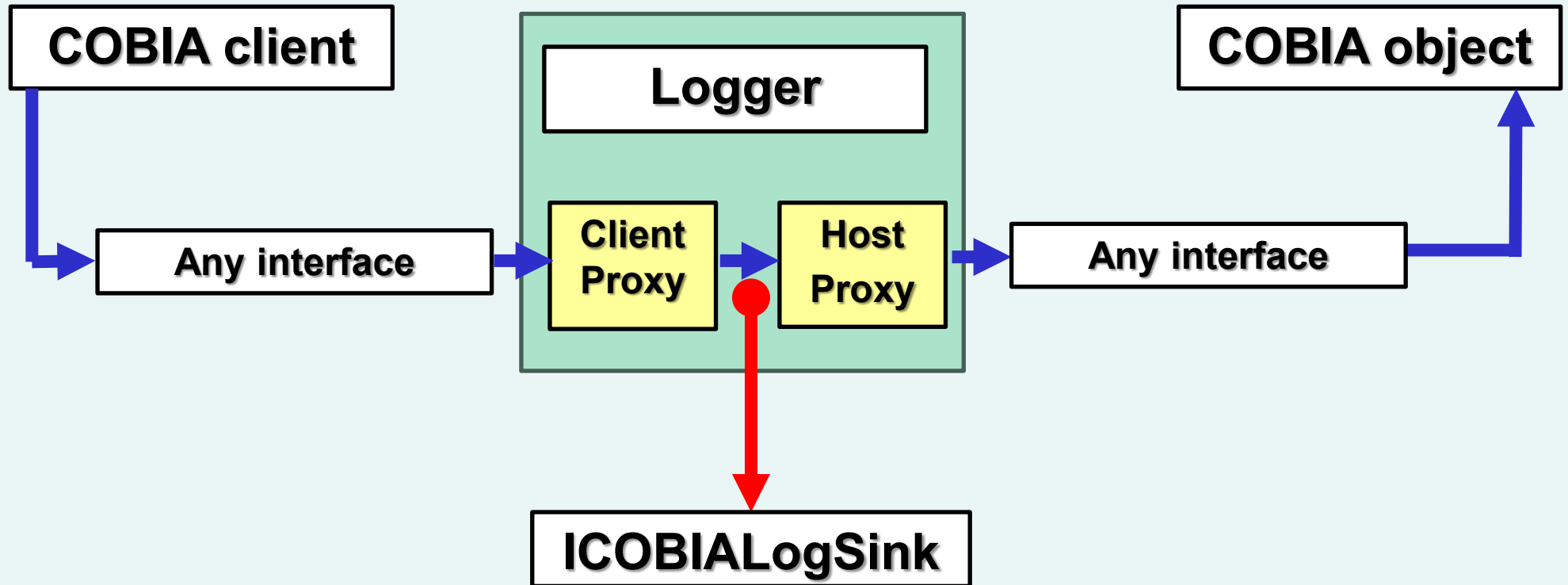
- Revision needed for default folders
- WindowID to be sorted
- Planning to distribute as Framework (Bundle)

Transport

- TCP/IP transport was demonstrated 2 years back
- TCP/IP is not secure – should be used behind a firewall only
- Secure transport should be designed
- M&T has been looking into SSH
 - Pre-installed on all systems
 - Provides all necessary means of authorization
- Needs a business case



COBIA Logging



Closing remarks

- COBIA Phase III code is now main branch**
- COBIA 1.2.1.1 beta is released**
- Soon to be followed by COBIA 1.2.1.2**
- The back-bones of COBIA are complete**
- COBIA is ready production use**
- COBIA provides a pathway for next CAPE-OPEN version**
- COM interop will remain**
- COM interfaces look slightly different**
- Existing COM functions can be re-used**

Closing remarks

- COBIA is free for use and redistribution
- CO-LaN recommends COBIA for new projects
- Redistribution should use CO-LaN provided binaries
- COBIA source code available to CO-LaN members
- Documentation and how-tos are sparse
- AmsterCHEM's COBIA Visual Studio wizard contains a step-by-step guide on creating a Unit Operation
- All code generation can be done from command line with COBIA tools (no third party tools needed)