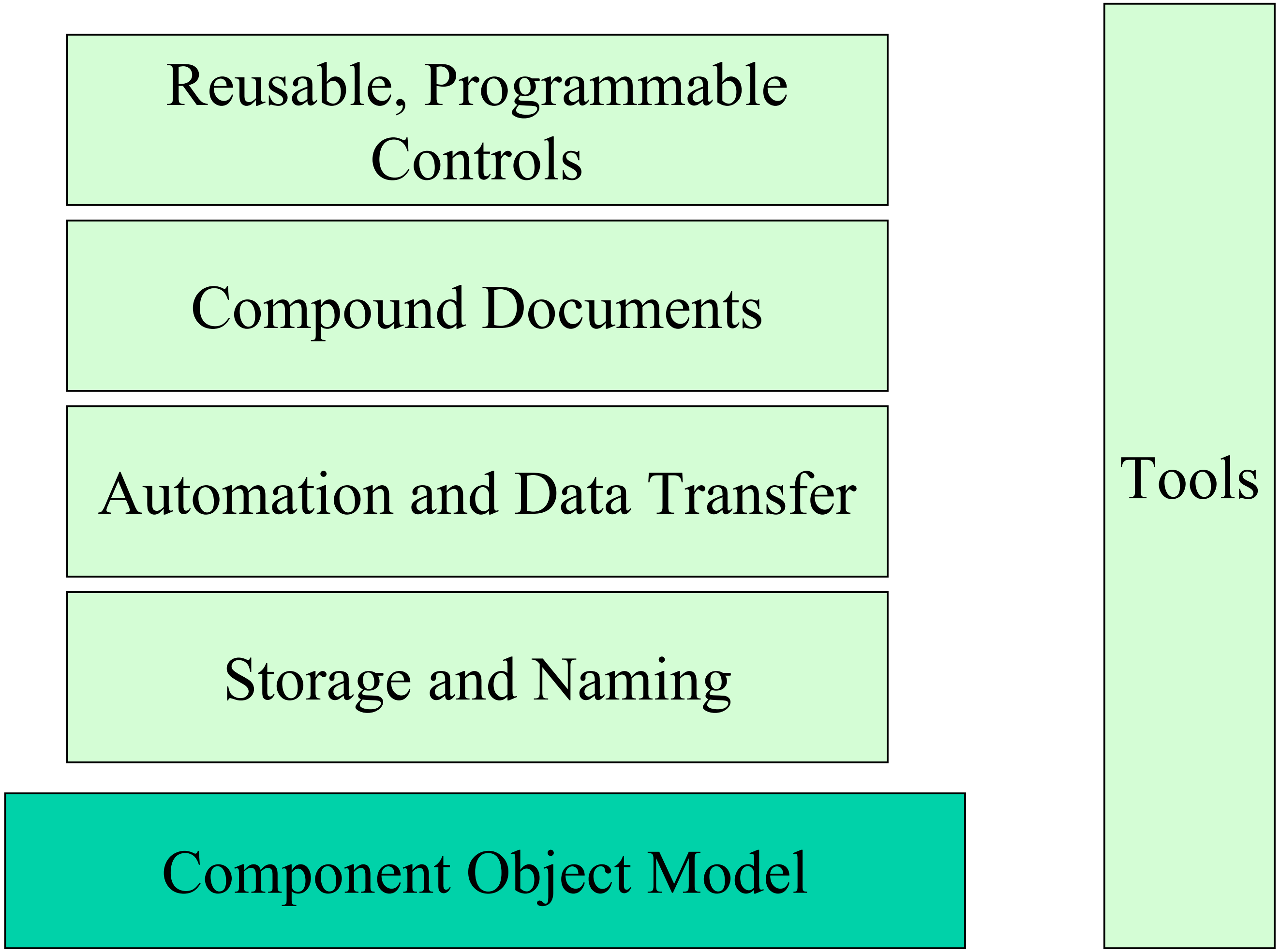
COM

To know new things,
learn by studying the old.

COM 1994


Reusable, Programmable Controls
Compound Documents
Automation and Data Transfer
Storage and Naming
Component Object Model
Tools

Bible

The Component Object Model Specification

Draft Version 0.9, October 24, 1995

Microsoft Corporation and Digital Equipment Corporation

今はどこにあるかわかんない。*(MSDN2001* にはあるけど*)*

Binary Standard

```
// unknwn.h
typedef struct IUnknownVtbl
{
    BEGIN_INTERFACE
    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *QueryInterface )( // 参照カウンター ++
            IUnknown __RPC_FAR * This,
            /* [in] */ REFIID riid,
            /* [iid_is][out] */ void __RPC_FAR *__RPC_FAR *ppvObject);
    ULONG ( STDMETHODCALLTYPE __RPC_FAR *AddRef )(  //参照カウンター ++
            IUnknown __RPC_FAR * This);
    ULONG ( STDMETHODCALLTYPE __RPC_FAR *Release )(  // 参照カウンター --
            IUnknown __RPC_FAR * This);
    END_INTERFACE
} IUnknownVtbl;

interface IUnknown
{
    CONST_VTBL struct IUnknownVtbl __RPC_FAR *lpVtbl;
};
```

interface

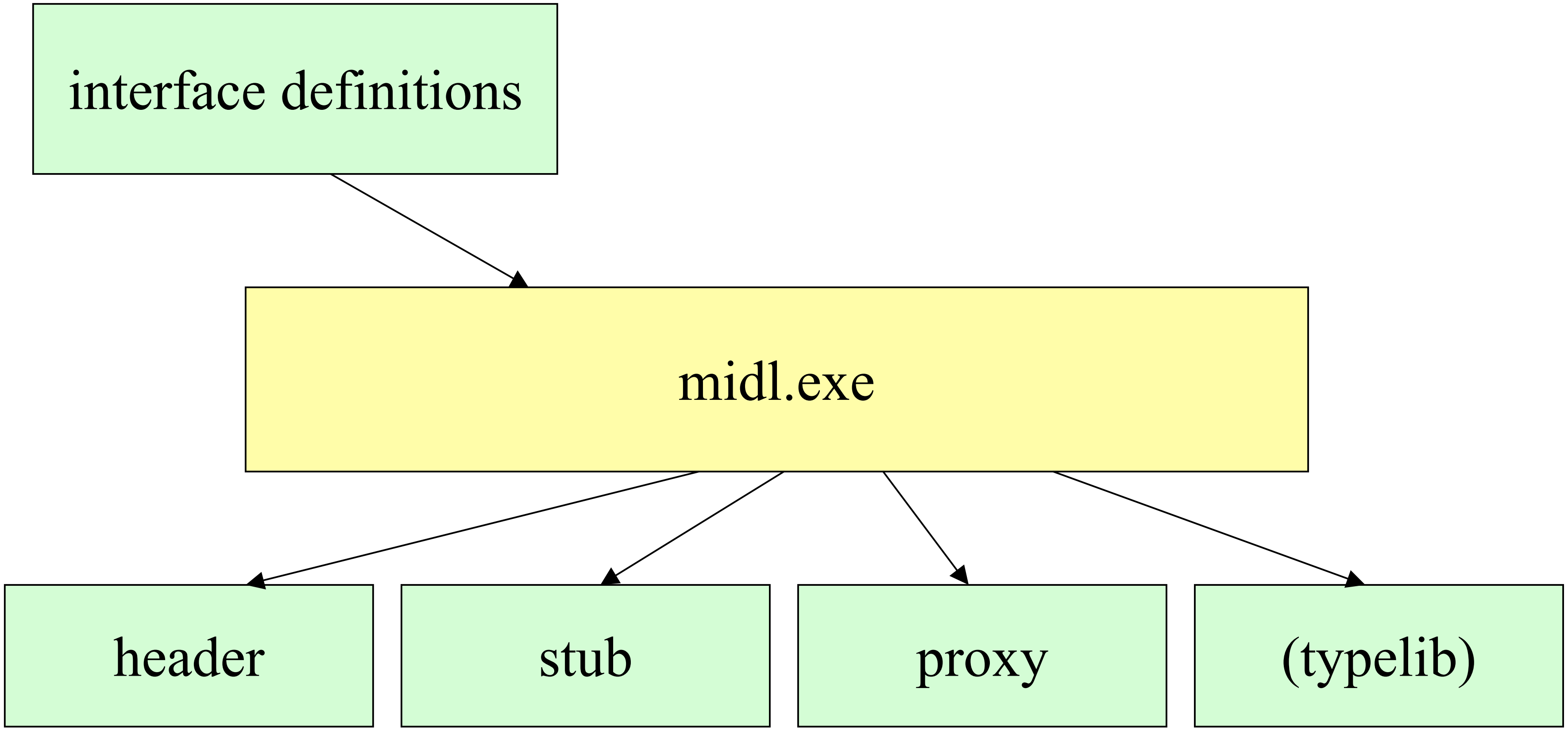
- **An interface is not a class**
- **An interface is not a component object**
- **Clients only interact with pointers to interfaces.**
- **Component objects can implement multiple interfaces**
- **Interfaces are strongly typed.**
- **Interfaces are immutable**

GUID

- globally unique ID

```
typedef struct GUID {
    DWORD Data1;
    WORD Data2;
    WORD Data3;
    BYTE Data4[8];
} GUID  ;
```

IDL



Memory Management Rules

- **in** parameter Allocated and freed by the caller.
- **out** parameter Allocated by the callee; freed by the caller.
- **in-out** parameter Initially allocated by the caller, then freed and re-allocated by the callee if necessary. As with out parameters, the caller is responsible for freeing the final returned value.

automation

- VARIANT
- BSTR
- SAFEARRAY
- IDispatch

- TypeInfo

IDispatch

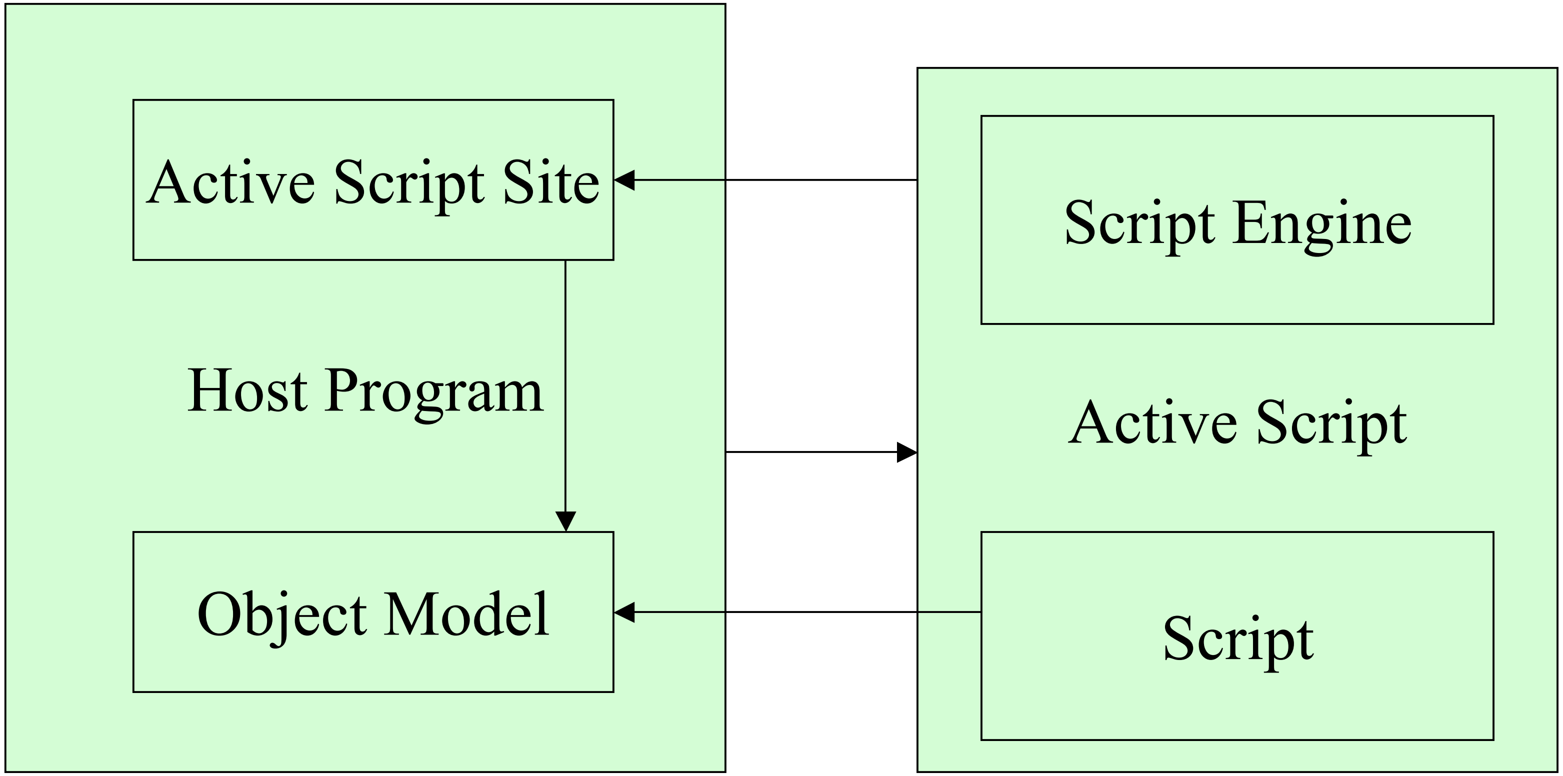
```
class IDispatch < IUnknown
  def getTypeInfoCount
    # 0 or 1, type information count
  end
  def getTypeInfo(localeid)
    # get type information
  end
  def getIDsOfNames(name, localeid)
    # find dispids by methodname
  end
  def invoke(dispid, params)
    # call the method
  end
end
```

Dual

```
interface IFoo : IDispatch {
    HRESULT Foo([in]long val, [out, retval]long* ret);
};

dispinterface Foo {
  methods:
   [id(DISPID_FOO)] long Foo(long val);
};
```

ActiveScript



Threading Model

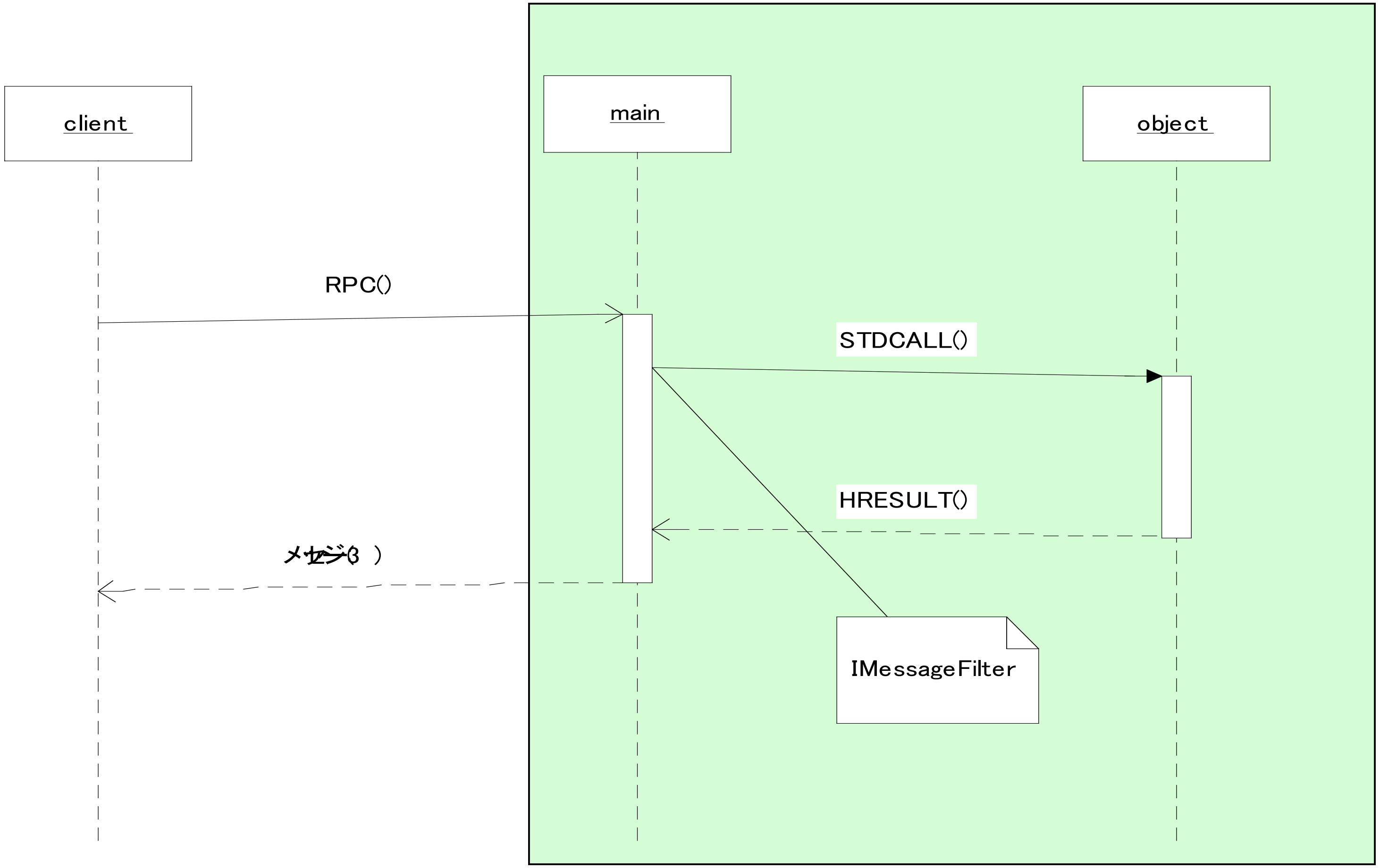
- ST　~NT 3.5
- STA/MTA　NT3.51（Windows95）~
- NTA　（Windows2000）~

- Single, Apartment, Free, Both, Neutral Obj.
- GIT

Marshal

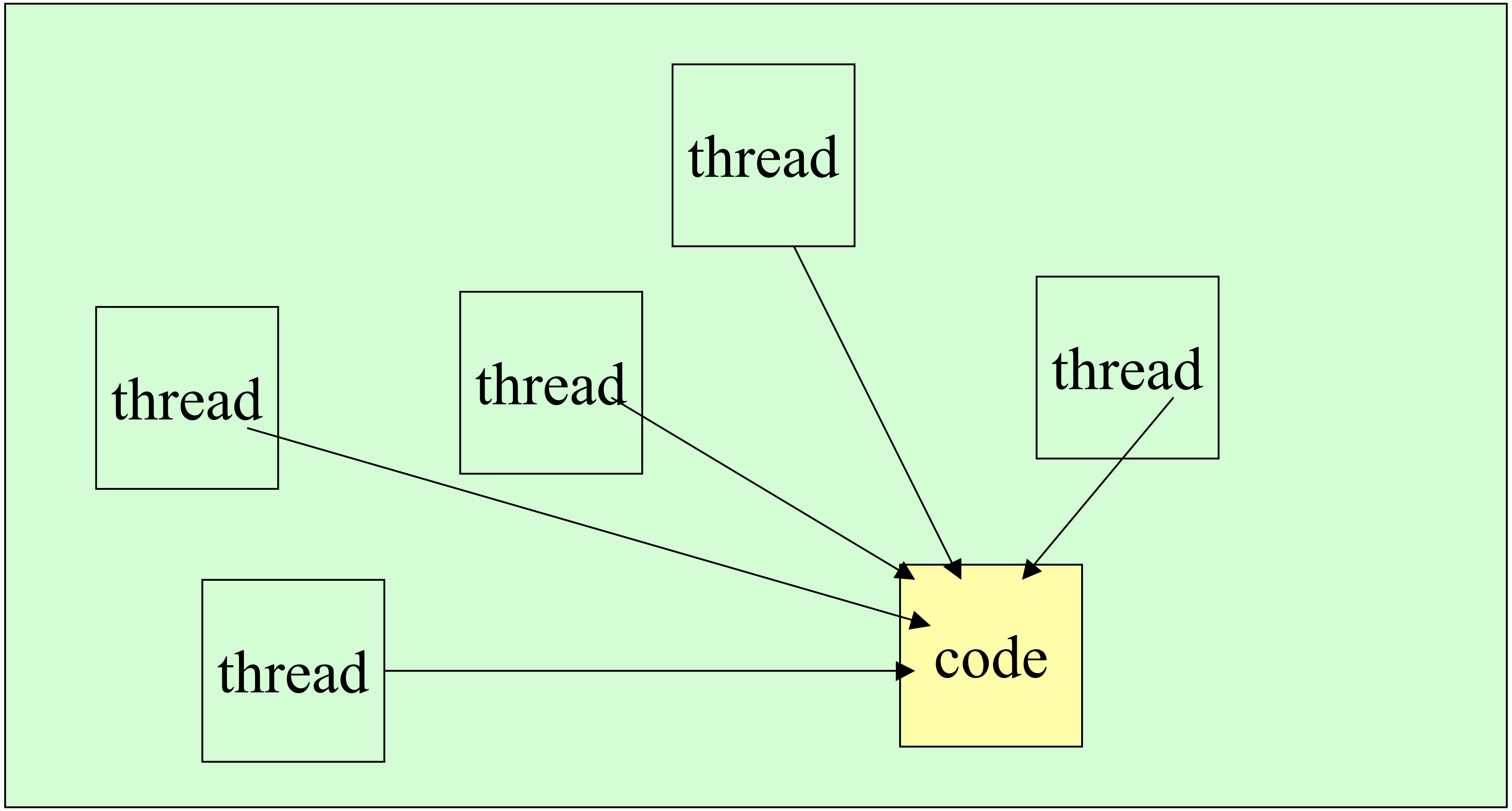
- CoMarshalInterThreadInterfaceInStream
- CoGetInterfaceAndReleaseStream
- CoCreateFreeThreadedMarshaler
- IMarshal (for Marshal By Value)

Message Pump (ST、STA）



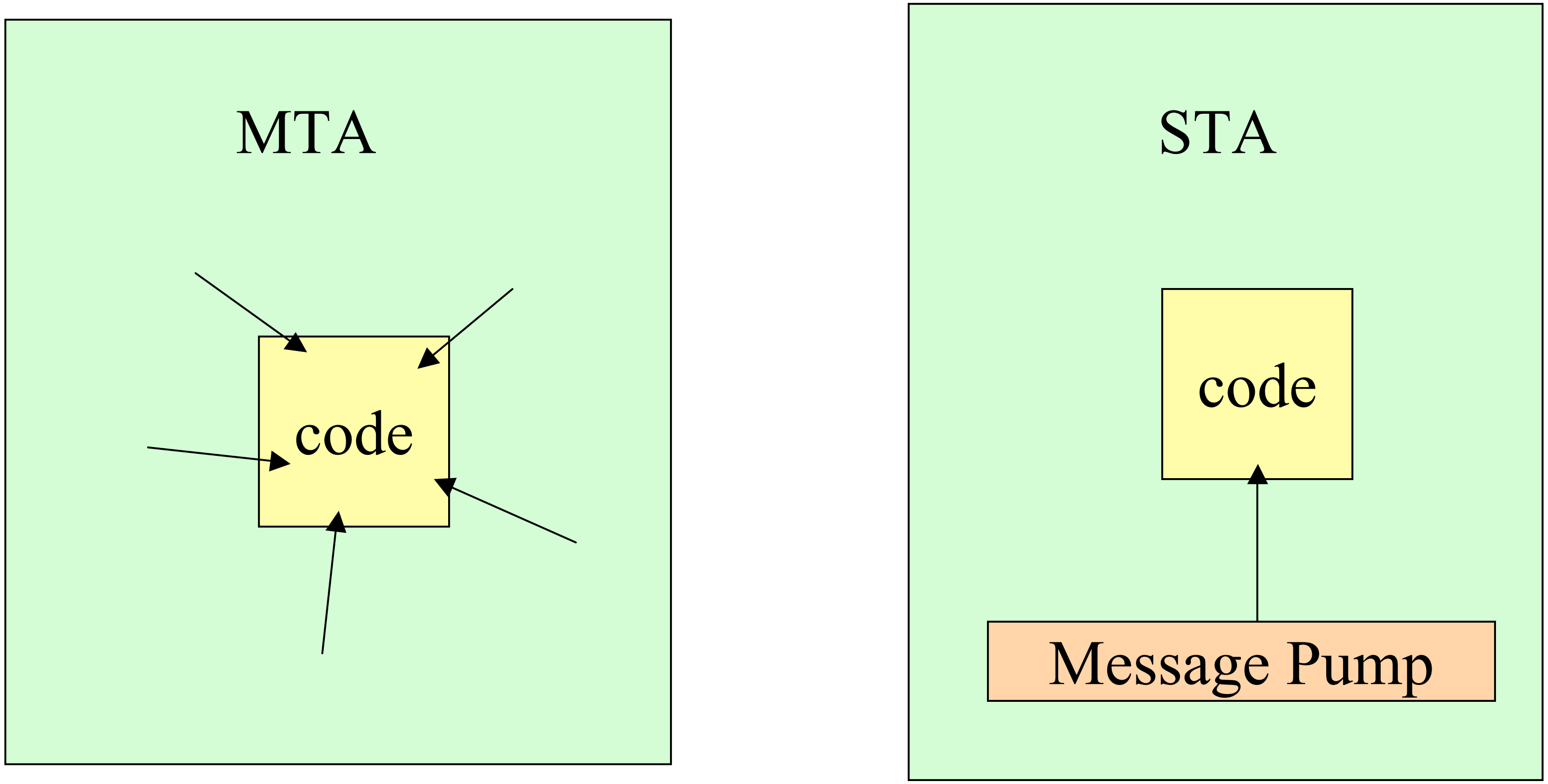
Beware reentrancy !

MTA



Beware race condition !

BOTH



Bewear composited objects !

NTA

- Sorry, I’ve quit the com front.