Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (01223) 515010**
INTERNATIONAL: **+44 1223 515010**
FAX: **+44 1223 359779**
E-MAIL: **apm@ansa.co.uk**

ANSA Phase III

A Lightweight Distributed Processing Infrastructure

Dave Otway

Abstract

There is an increasing need to add real-time capability to distributed systems [APM.1581].

But the existing distributed processing infrastructures have been designed to optimise resource usage and to be very comprehensive in their functionality. This makes them too slow and cumbersome to support real-time applications.

Before adding real-time capabilities, it is necessary to build a flexible, lightweight framework for distributed processing infrastructures that can be tailored precisely to application requirements, but which can still interoperate with other infrastructures.

APM.1580.02 **Approved** 13th October 1995
Briefing Note

Distribution:
Supersedes:
Superseded by:

A Lightweight Distributed Processing Infrastructure

Dave Otway

Contents

- Requirements, Background, Objectives, Plan
- Computational API
- Engineering API
- Stub generator
- DIMMA nucleus

Requirements for a Real-Time Distributed Infrastructure

- **Add real-time capabilities to the ANSA/ODP architecture**
  - **preserving its ability to cope with: federation, heterogeneity, scaling**
- **Provide interoperability between real-time and non real-time objects and between real-time and non real-time distributed infrastructures**
- **Provide a flexible modular framework that can be used to build optimised infrastructures for specialised applications**
- **Provide real-time guarantees in an asynchronous distributed system**
  - **predictable islands in an unpredictable sea**

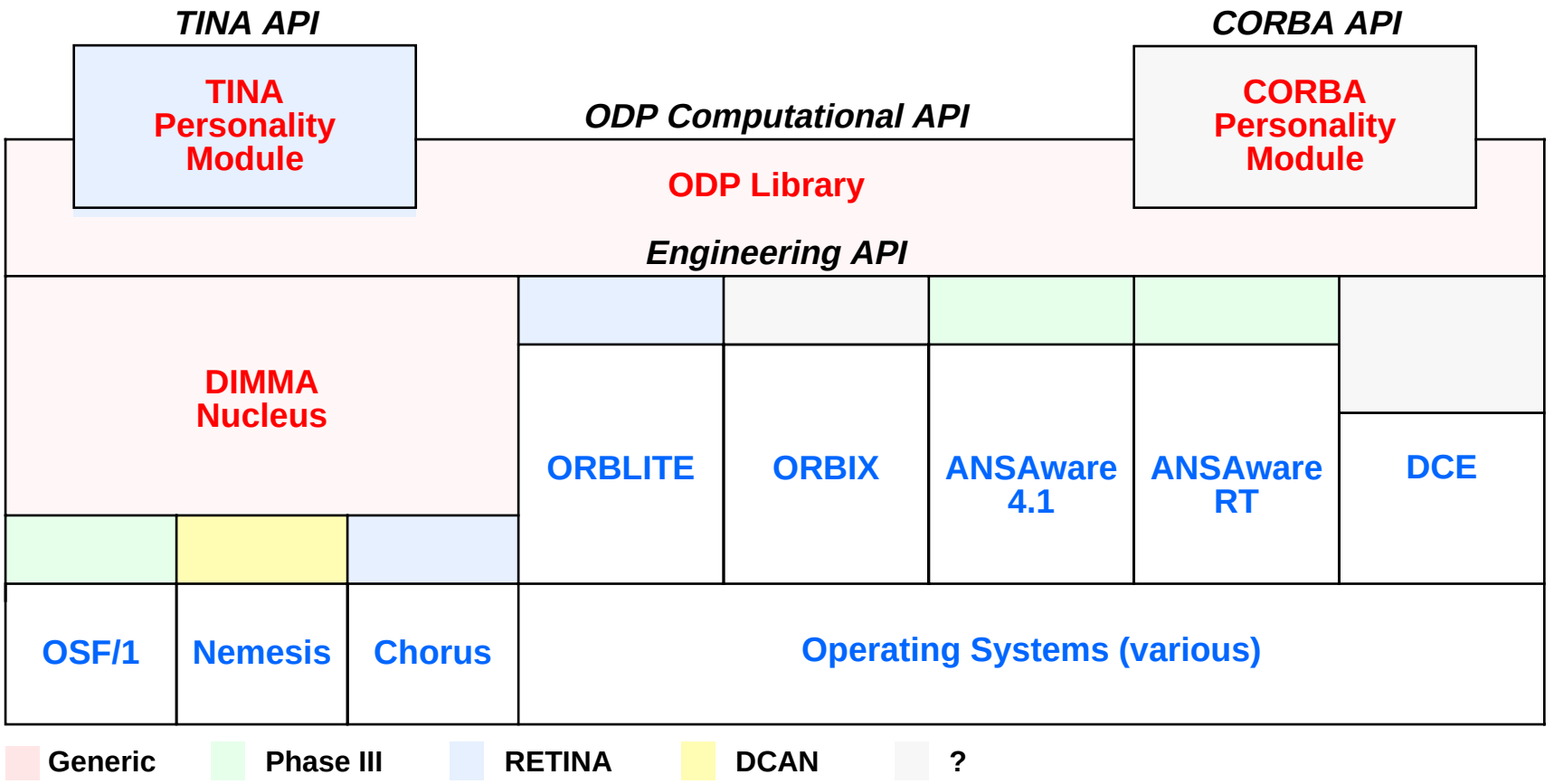
Background

- ANSAware is not lightweight enough
  - designed for efficient resource usage [too much sharing & multiplexing]
- ODP Computational model is now an ISO standard
  - this has exactly the right semantics
- CORBA is becoming the “standard” product
  - but we need to remain vendor neutral
  - and CORBA products also suffer from the same problems as ANSAware
- there is a requirement for a lightweight ORB with Real-Time capabilities
- C++ provides the functionality needed for distributed programming

Objectives

- develop a portable ODP conforming computational API in C++
  - with a portable engineering API
  - which is protocol and platform independent
- develop a modular lightweight distributed infrastructure framework
  - with generic infrastructure components
- develop a flexible set of multi-IDL stub generators and IDL translators
  - based around a conformance type checking AST
- understand, explain and map the semantic differences between different APIs, IDLs and Distributed Infrastructures

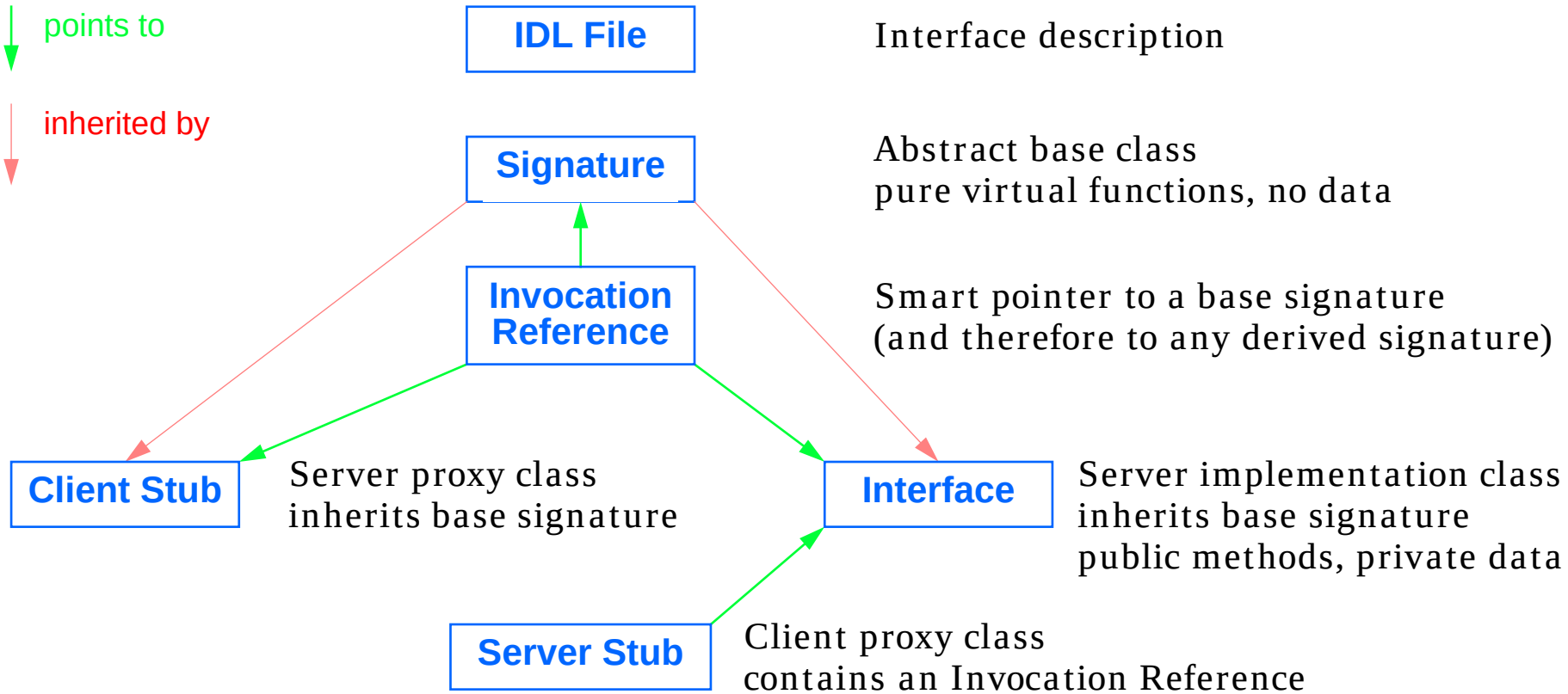
Master Plan


TINA API
CORBA API
TINA Personality Module
ODP Computational API
CORBA Personality Module
ODP Library
Engineering API
DIMMA Nucleus
ORBLITE
ORBIX
ANSAware 4.1
ANSAware RT
DCE
OSF/1
Nemesis
Chorus
Operating Systems (various)
Generic
Phase III
RETINA
DCAN
?

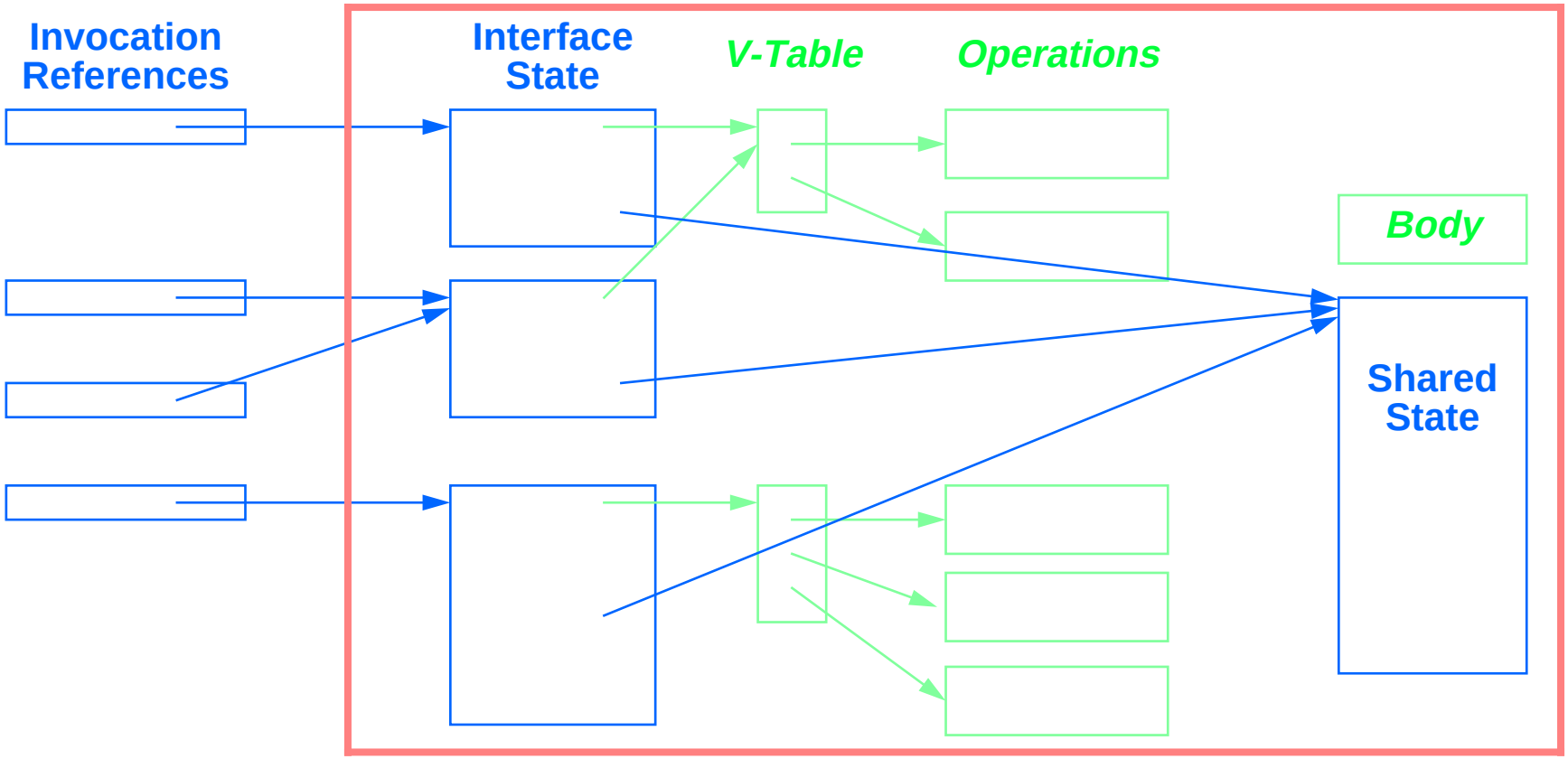
Computational API

- **Stage 1:** objects, interfaces, multiple results, basic types, the type "Any"; signatures, invocation references, named terminations, local garbage collection, (hand coded) trader client stub
- [stage 1+ sequences implemented over DIMMA and ANSAware 4.1]
- stage 2: structured types, threads, service withdrawal
- stage 3: streams, synchronous programming, explicit binding, QoS
- stage 4: preprocessor

Interface component relationships



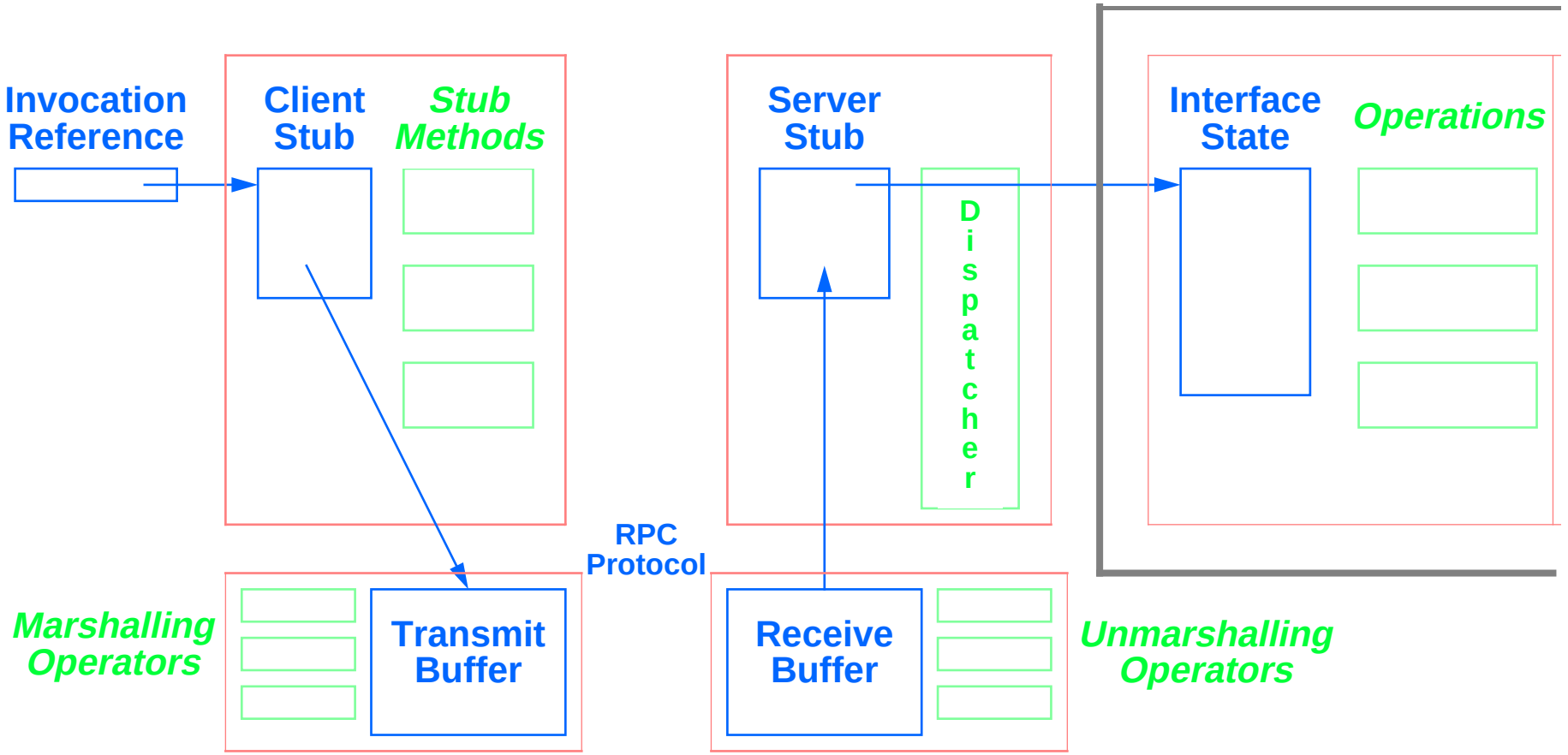
Implementing an ODP Object in C++



Engineering API

- supports protocol and platform independent stubs with:
- client stub and server stub base classes
  - derived client and server stubs produced by the stub generator
- transmitter and receiver base classes with virtual marshalling operators for the basic types
  - derived buffers have to be implemented for each RPC protocol
  - binder selects appropriate buffer for the client
  - protocol selects appropriate buffer for the server
- templates for generating sequence and array mashalling operators
  - structure and union mashallers produced by the stub generator

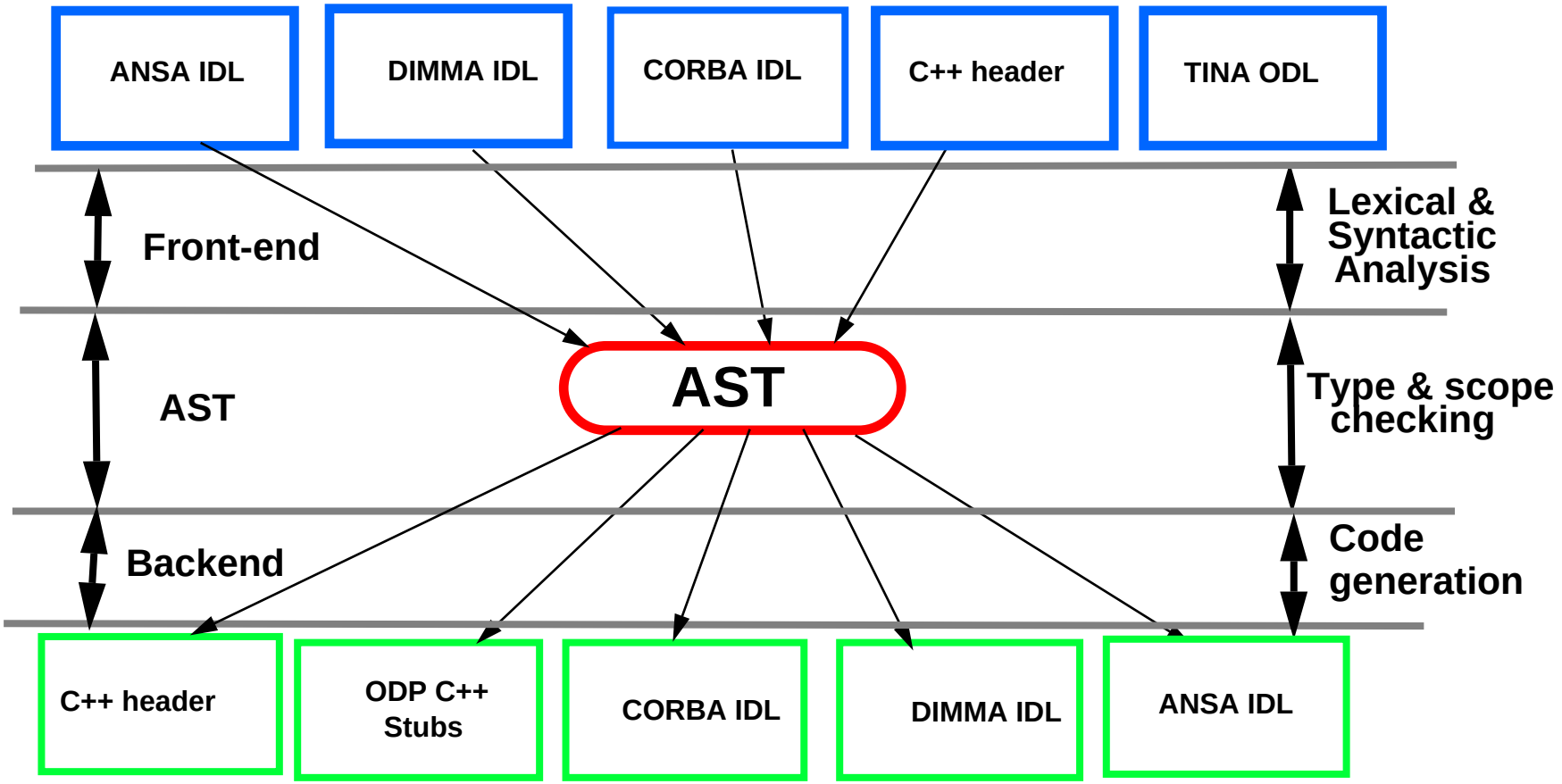
A Remote Invocation



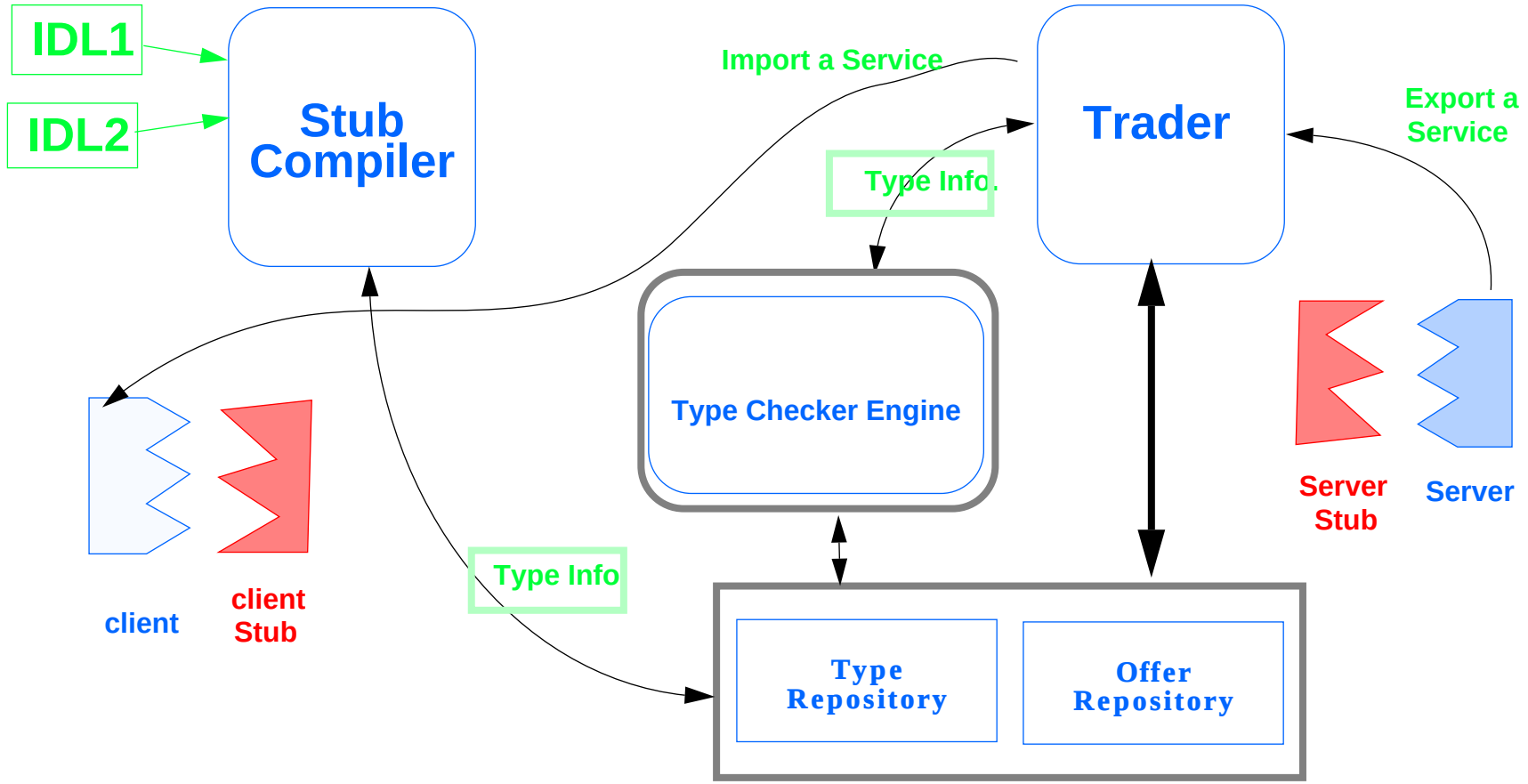
Stub Generator

- Support several input notations to describe service interfaces (CORBA IDL, C++ headers, ANSA IDL, DIMMA IDL, etc.)
- Support several runtime environments (transport, data formats, etc.).
- Generate headers, stubs and skeletons targeting a language mapping (e.g C++) and an runtime environment (e.g. ODP runtime).
- Translate an input IDL notation to another equivalent IDL notation (e.g DIMMA IDL -> CORBA IDL), flagging incompatibilities.
- Perform Type and Scope checking.

Stub Generator Master Plan



Type-Safe Trading



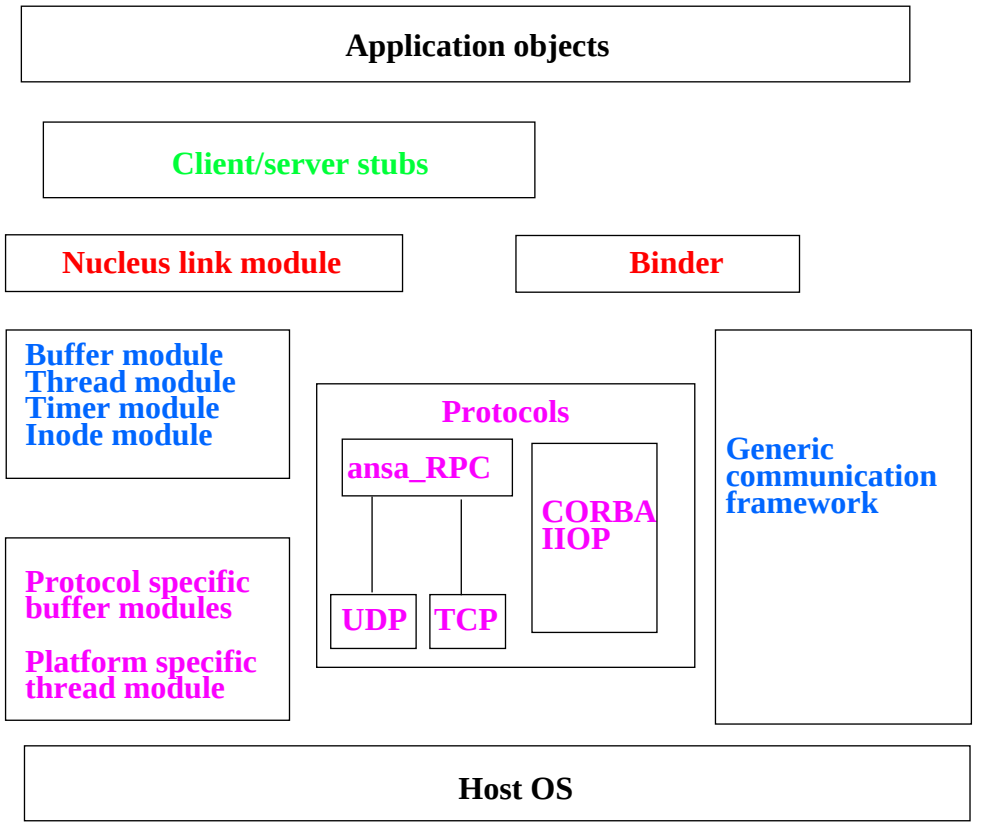
Why a framework ORB?

- real world needs many ORBs
  - high performance, downsize, scalable
  - transactional, reliable, real-time, multimedia
- technology advances - must be reflected in future ORBs
  - more functional scheduling support
  - more powerful communication support
  - other tools
- connectivity
  - many protocols, many existing ORBs, alternative object APIs

-> a microkernel ORB!

Modularity in the Nucleus

- communication - a generic communication framework
- processing
  - many threading schemes (multi-threading, non-threaded, real-time threading)
- memory management
  - many buffering schemes (eager, continuous, linked list)
  - many Protocol Data Units (PDUs)
  - many presentation protocols
- event processing
  - synchronous
  - asynchronous
  - half synchronous/asynchronous



Nucleus Structure

Related Activities

- **RETINA**
  - TINA computational API, engineering API, DIMMA Nucleus, Chorus and ORBLITE Adapters, stub generator
- **DCAN**
  - ODP computational API, engineering API, DIMMA Nucleus, OSF/1 and Nemesis adapters, stub generator
- **other sponsors**
  - ODP computational API, engineering API, DIMMA Nucleus, stub generator
- **ICL**
  - DIMMA Nucleus, Stub generator, CORBA computational API