**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**

# MC  AST  presentation (07/95)

## Youcef Laribi

**Abstract**

These slides give an overview of the Stub compiler toolset work carried out as part of the DIMMA project. We first analyse the objectives targeted by this work and the functions supported or enabled by the resulting toolset. We then give an outline architecture of the toolset, underlying its important components and their interfacing to each other. Finally, we give an update status of the work achieved and the different components that are operational.

| | | |
|---|:---:|---|
| APM.1523.01 | **Approved** | 11th July 1995 |
| | Briefing Note | |

**Distribution:**
**Supersedes**:
**Superseded by**:

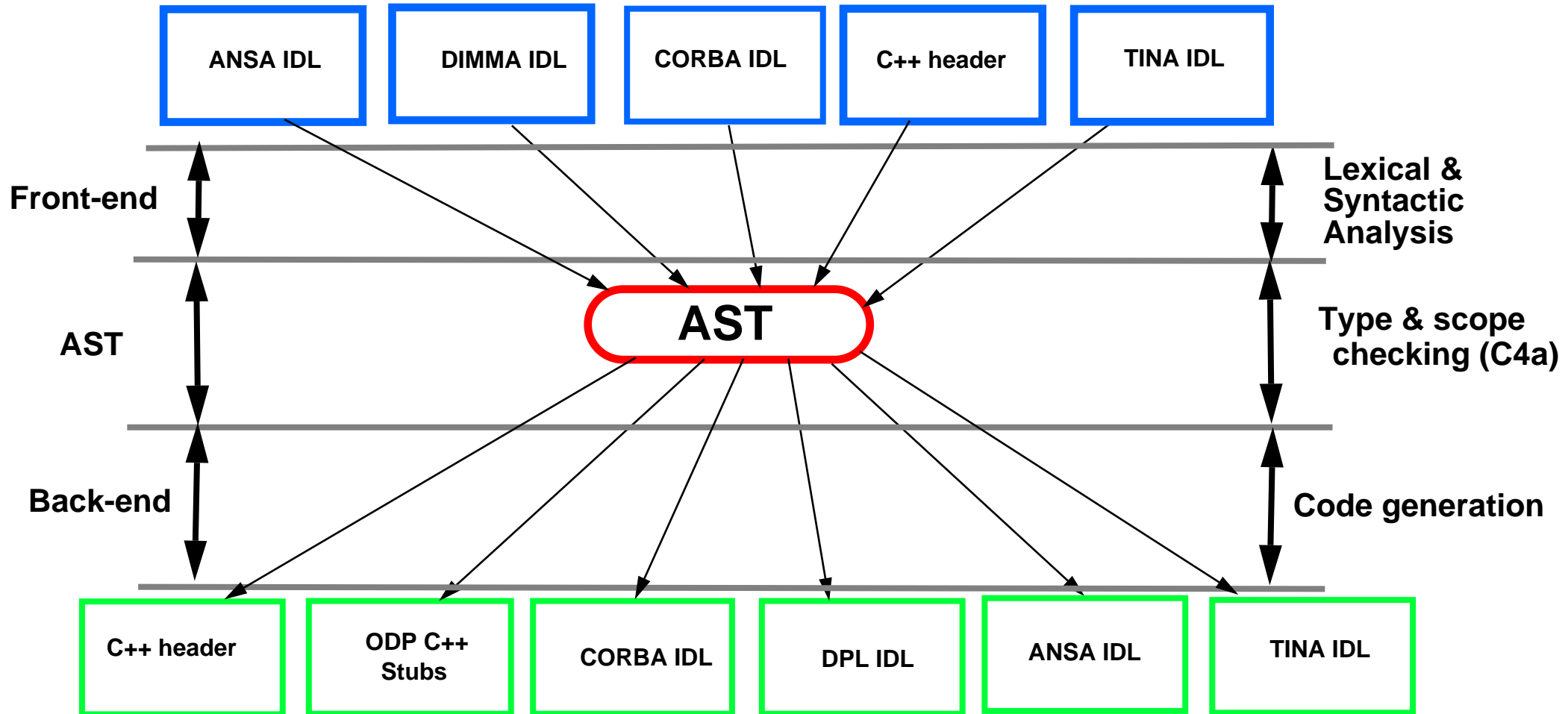# An AST-Based Stub Compiler

**Youcef Laribi (yl@ansa.co.uk)**
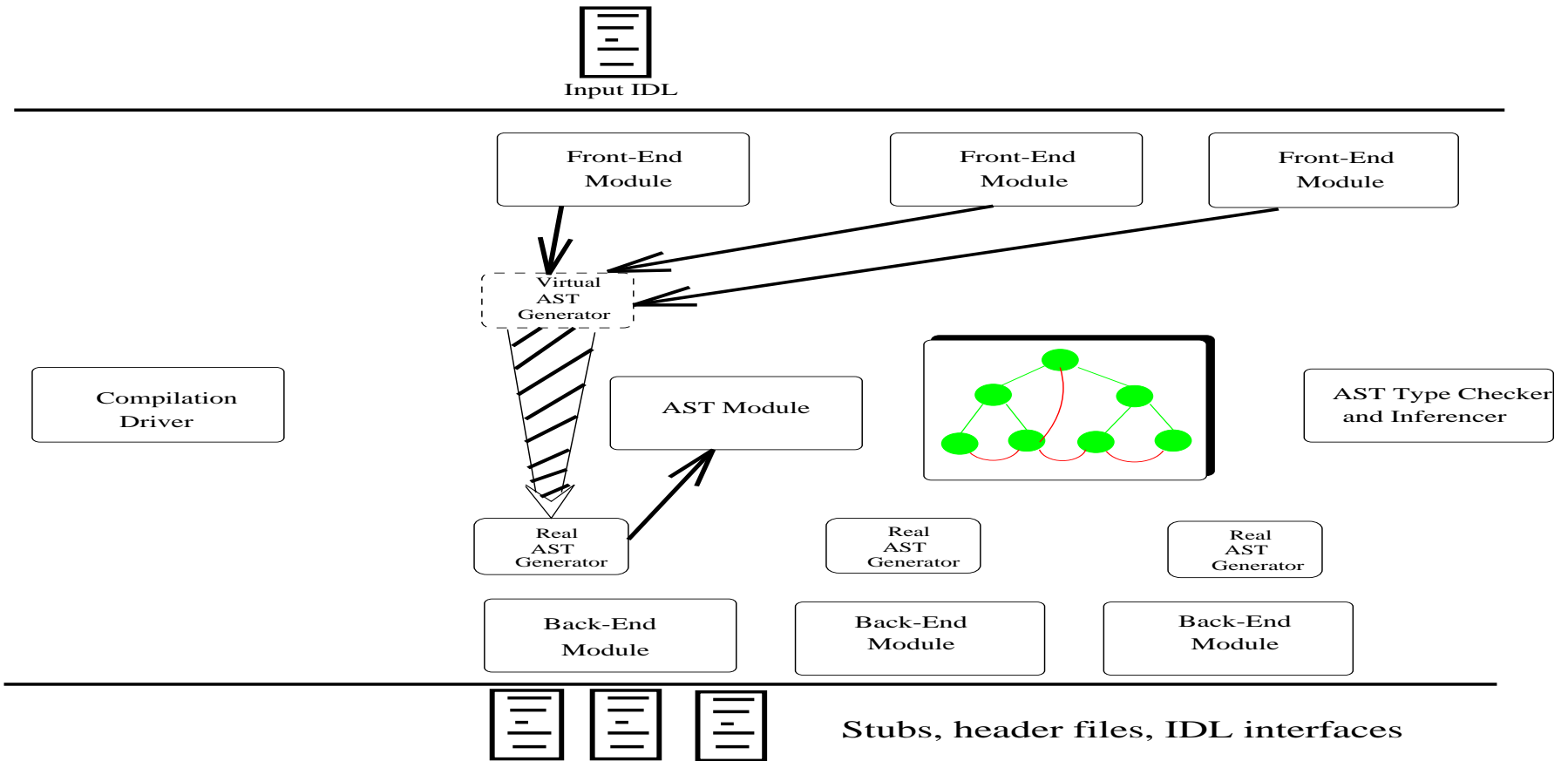
# AST-based IDL compiler

## Objectives:

- **Support several input notations to describe service interfaces (CORBA IDL, C++ headers, TINA IDL, etc).**

- **Generate headers, stubs and skeletons targetting a language mapping (e.g C++) and an engineering API (e.g DIMMA nucleus).**

- **Output the type information required for a type-safe trading.**

- **Translate an input IDL notation to another equivalent IDL notation (e.g DIMMA IDL -> CORBA IDL), flagging incompatibilities.**

# IDL Master Plan



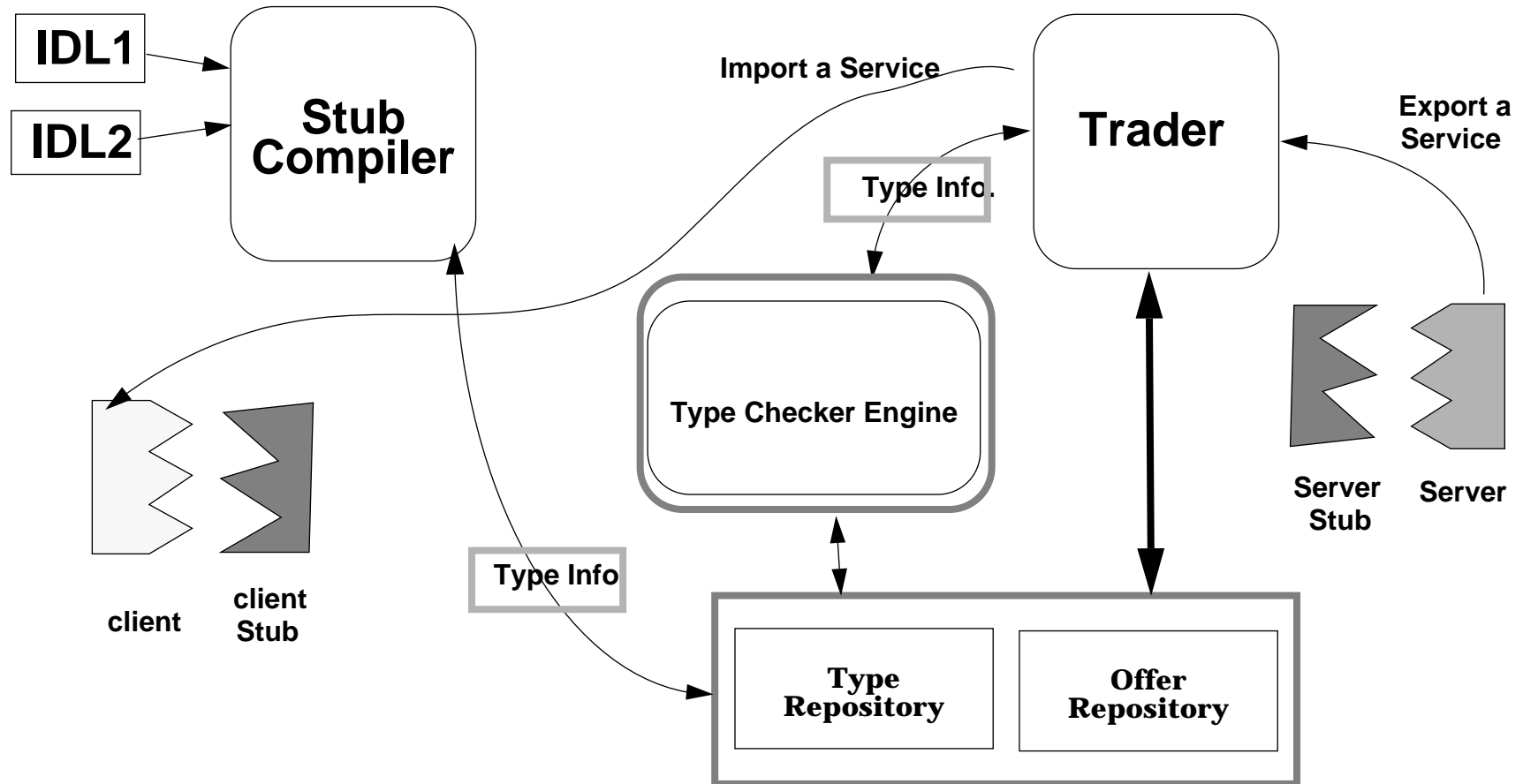ANSA IDL  DIMMA IDL  CORBA IDL  C++ header  TINA IDL

**Front-end**

**Lexical & Syntactic Analysis**

**AST**

**AST**

**Type & scope checking (C4a)**

**Back-end**

**Code generation**

C++ header  ODP C++ Stubs  CORBA IDL  DPL IDL  ANSA IDL  TINA IDL

3

# AST Tools architecture

**Input IDL**

Front-End Module

Front-End Module

Front-End Module

Virtual AST Generator

Compilation Driver

AST Module

AST Type Checker and Inferencer

Real AST Generator

Real AST Generator

Real AST Generator

Back-End Module

Back-End Module

Back-End Module

Stubs, header files, IDL interfaces

# Type-Safe Trading

**IDL1** → **Stub Compiler**

**IDL2** → **Stub Compiler**

**Import a Service**

**Trader**

**Export a Service**

**Type Info.**

**Type Checker Engine**

**client**   **client Stub**

**Type Info**

**Server Stub**   **Server**

**Type Repository**   **Offer Repository**

# Progress

- **Built a DIMMA IDL front-end (lexical, syntactic analysis and AST generation).**

- **Almost completed a C++ back-end which generates stubs for the ODP API.**

- **Enabled for the usage of the type checking engine.**

- **Working on the integration of the Sun's freely available CORBA IDL front-end to generate stubs for ODP API from CORBA IDL descriptions.**