



**Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom**

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (01223) 515010
+44 1223 515010
+44 1223 359779
apm@ansa.co.uk**

ANSA Phase III

PA1:(SYNCH ARCH) and PA2:(BINDING ARCH) sign-off slides

Dave Otway

Abstract

This presentation contains two requests for sign-off of work-items by the December 94 TC..

:PA1:(SYNCH ARCH) Documented in APM.1108.00.02: Streams and Signals

PA2:(BINDING ARCH) Documented in APM.1239.00.03: The ANSA Binding Model

It then documents some ideas on declarative binding for discussion..

APM.1372.01

Approved
Briefing Note

5th December 1994

Distribution:

Supersedes:

Superseded by:



Binding, Streams and Signals

Sign-off and Discussion

Dave Otway

djo@ansa.co.uk



Contents

PA1:(SYNCH ARCH) sign-off

PA2:(BINDING ARCH) sign-off

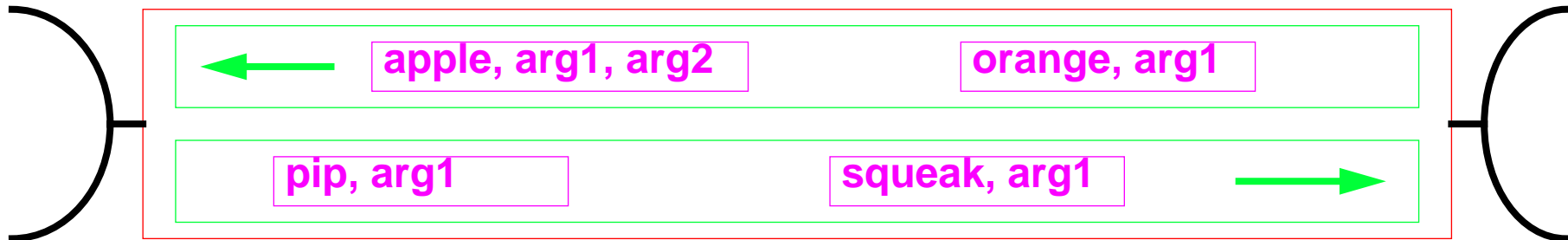
Declarative Binding



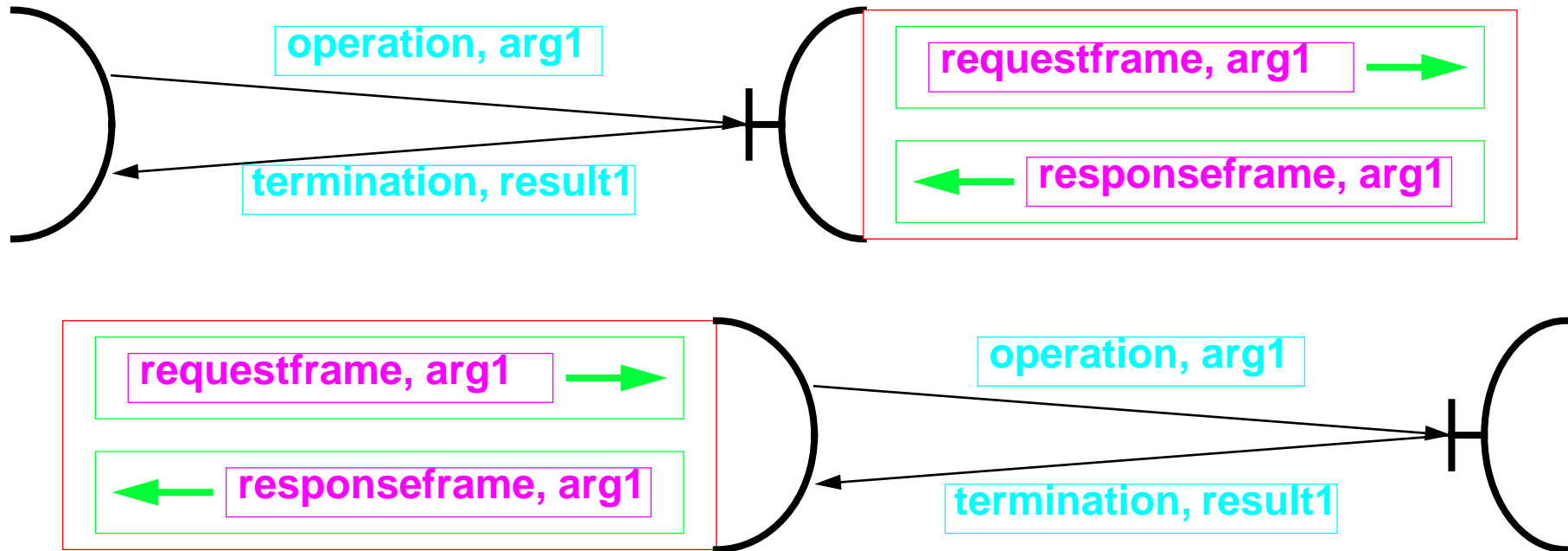
Streams and Signals

- **Document APM.1108.00.02: “Streams and Signals”**
- **Was workpackage PA1: (SYNCH ARCH)**
 - in plan in operation while work was being done
- **Now parts (b) and (c) of workpackage C2**
 - in current plan
- **C2: Architecture for distributed interactive multi-media**
 - **b) interfaces for interaction with a wide range of multi-media sources (signal interfaces)**
 - **c) synchronization of activities driven by streams and signals**

Streams



- **A stream has a set of flows**
- **A flow has a set of frames (or signals) and a direction**
- **A frame has a name and a set of typed arguments**
- **Streams are typed and can be conformance type checked**
- **Frames are transmitted by non-blocking writes and read by blocking reads**



- A stream interface can be programmed to use or provide an operational interface
- But relies on the programmer correctly implementing the operation invocation semantics



Synchronous Programming

- Synchronous signals are just synchronised frames
- Program execution is divided into discrete instants
 - signals present at the start of an instant are processed during the instant
- Determinism is achieved by bounded execution paths, elimination of internal signals by thread serialisation, and scheduling guarantees
- Time synchronisation is done by synchronising data & clock signals
- Extra synchronous programming constructs:
 - to test for the presence of signals
 - to wait for signals
 - to interrupt a series of reactions and execute an alternative series



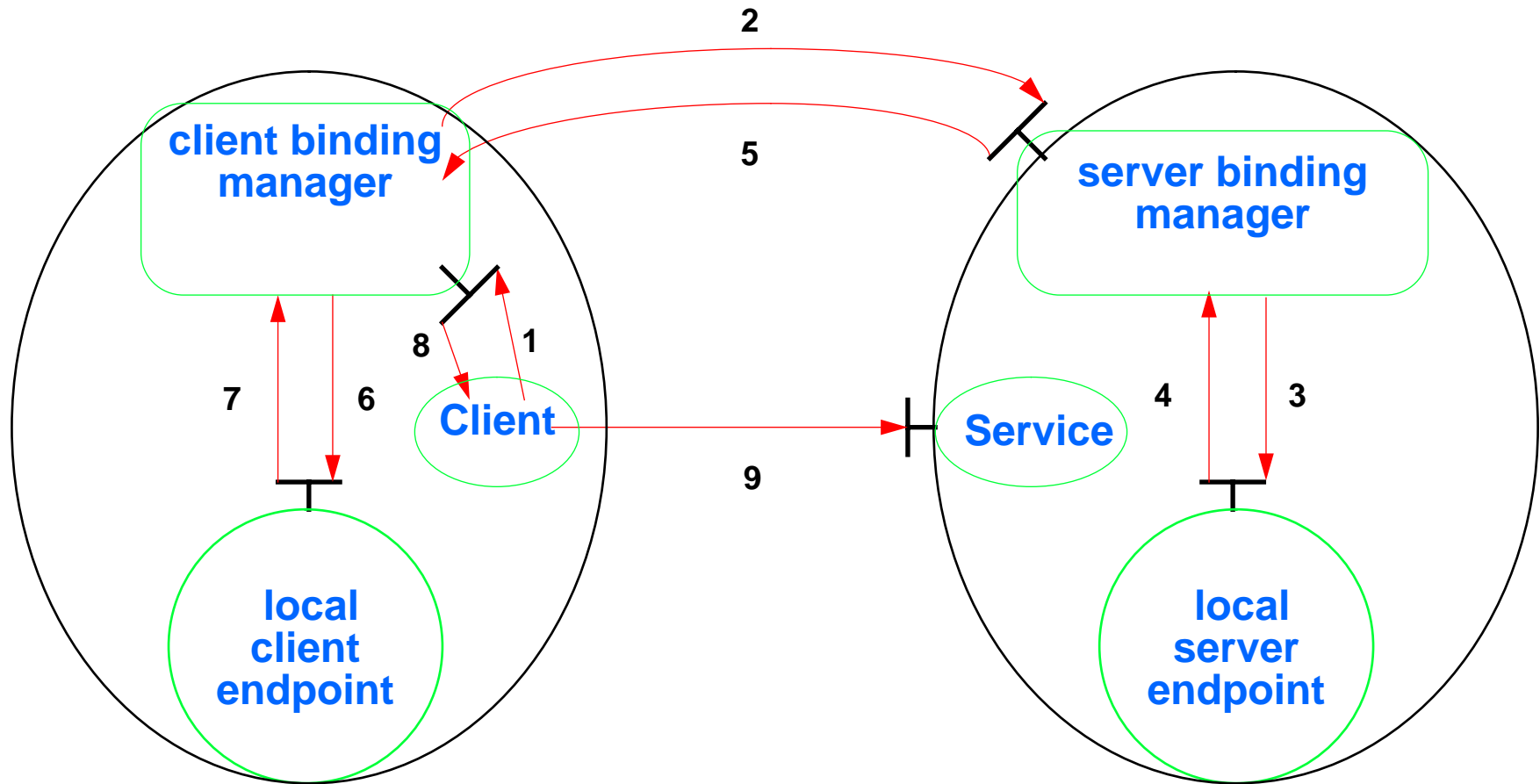
Explicit Binding

- Document APM.1239.00.03: “The ANSA Binding Model”
- Was workpackage PA2: (BINDING ARCH)
 - in plan in operation while work was being done
- Now part (a) of workpackage C2
 - in current plan
- C2: Architecture for distributed interactive multi-media
 - a) explicit binding of interfaces, including streams, including end-to-end definition of QoS
- Also documents the implicit binding engineering



Explicit Binding

- **Generate a type specific local binder for each end of a binding**
 - for client and server of an operational interface and both ends of a stream
- **Perform all binding management by application program (libraries)**
 - with the benefit of all distribution transparencies and tools
 - uses an implicit binding
- **Each local binder is then requested to construct its end of the binding**
- **Provide type safe QoS specification, monitoring and control interfaces, multi-channel and multi-party bindings via matching attribute and engineering libraries**
- **Does not require management protocols for explicit binding to be provided in the comms technology**





Declarative Binding

- **APM.1239 describes imperative binding mechanisms**
- **These are required in any case**
- **So they are the highest priority and should be implemented first**
- **But would a more declarative approach**
 - **improve the programming interface ?**
 - **enable use of higher level tools ?**



QoS Attributes

- **on type expressions**
 - **client specified QoS -- whenever and however a binding is made**
- **on interfaces**
 - **server specified QoS -- whenever and however a binding is made**
- **These require dynamic QoS conformance checks**
- **in IDL files**
 - **fixed joint QoS specification**
- **on blocks**
 - **scoped and nested default QoS specifications**
 - **unless specifically overridden**



Declarative specification of bind time

- **Establish all bindings used in a block before entering it**
 - what about bindings declared inside a block ?
 - can these always be evaluated before execution ?
 - what happens if an attempt to establish a binding fails ?
- **Dis-establish all bindings established in a block on exit**
 - including premature exit via named terminations (exceptions) ?
 - or should these be dealt with by the handler ?
 - implicitly or explicitly ?
- **Should binding control be part of a more generic mechanism for declarative control of resource allocation ?**
- **Are there other ways of grouping bindings into an all-or-nothing set ?**



Current Position

- Declarative binding seems like a good idea
- But implementing imperative explicit binding has higher priority
- There are no plans for work on declarative binding
- So we are recording a useful extension to the architecture
 - in case effort and priorities enable us to pursue it in the future