

SAM 98

Making Frameworks using SDL

Birger Møller-Pedersen ERICSSON
Rolv Bræk SINTEF



A framework is:

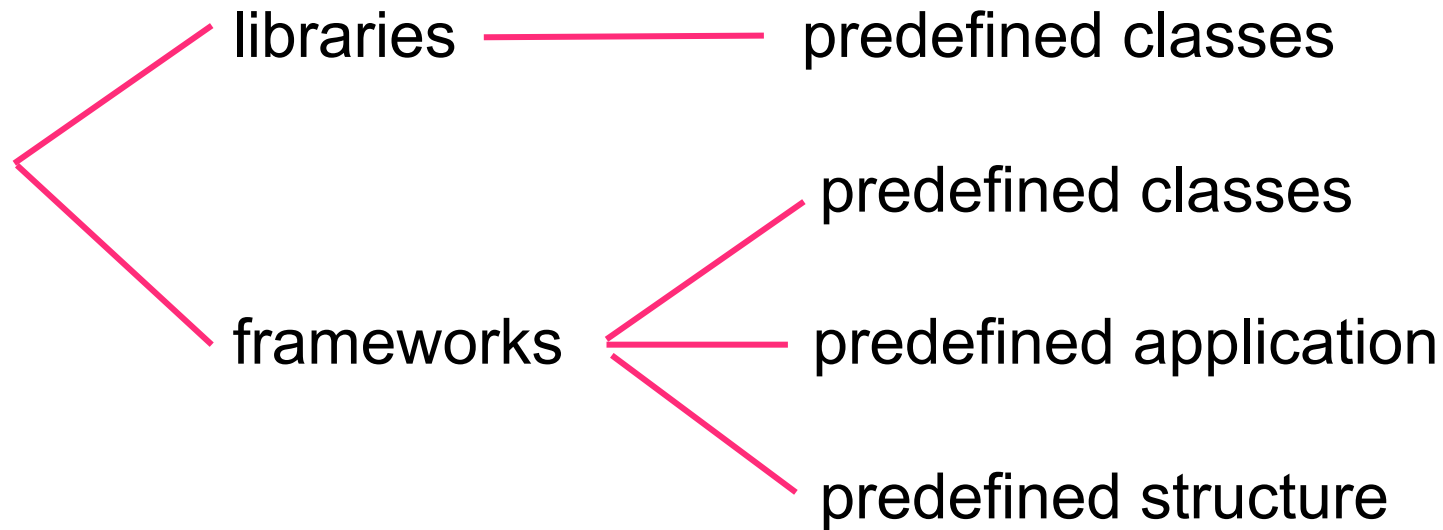
The Free On-line Dictionary of Computing :

“In object-oriented systems,
a set of classes that embodies an abstract design for
solutions to a number of related problems.”

From a tutorial:

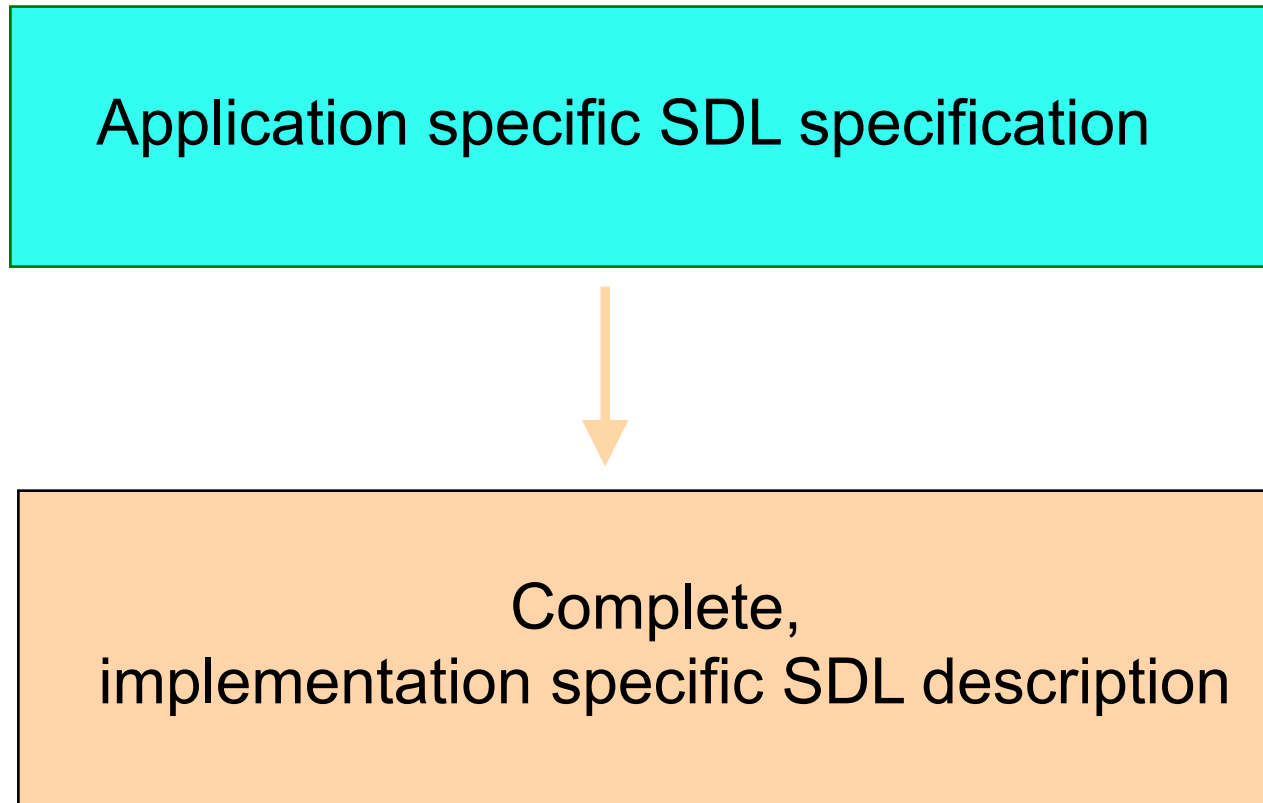
“Frameworks are reusable designs for an application or a subsystem
expressed as a set of classes and
the way that instances of these classes collaborate.”

Libraries and Frameworks

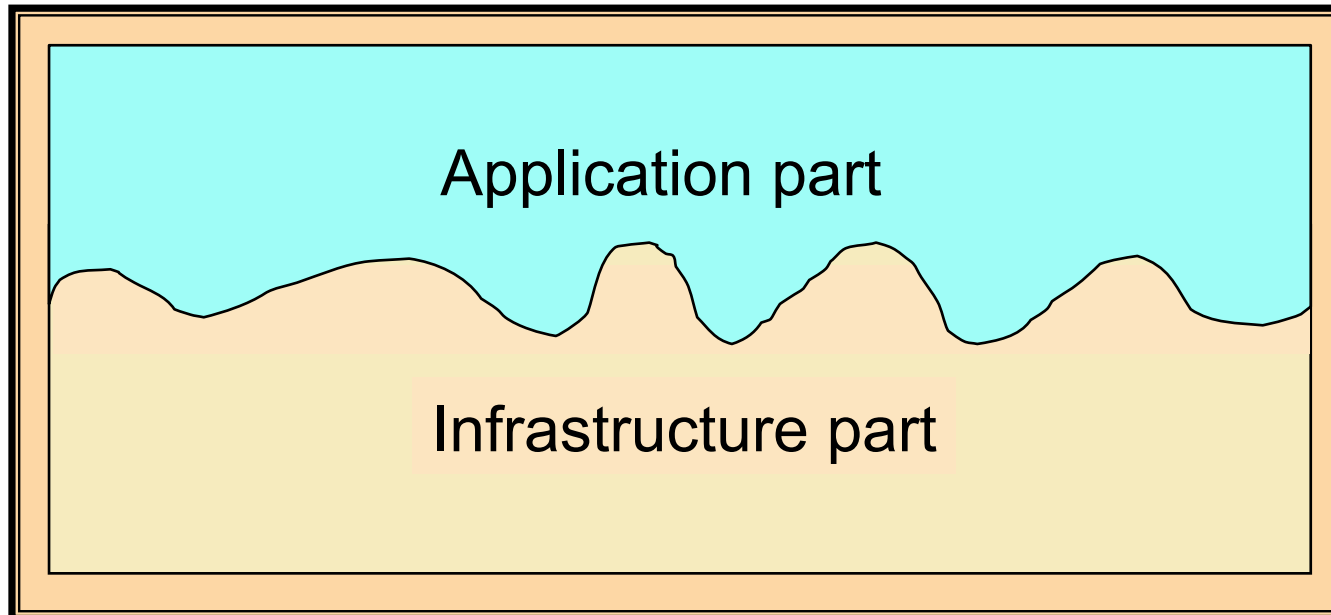


In SDL: system types
with predefined instance structure

Two SDL models



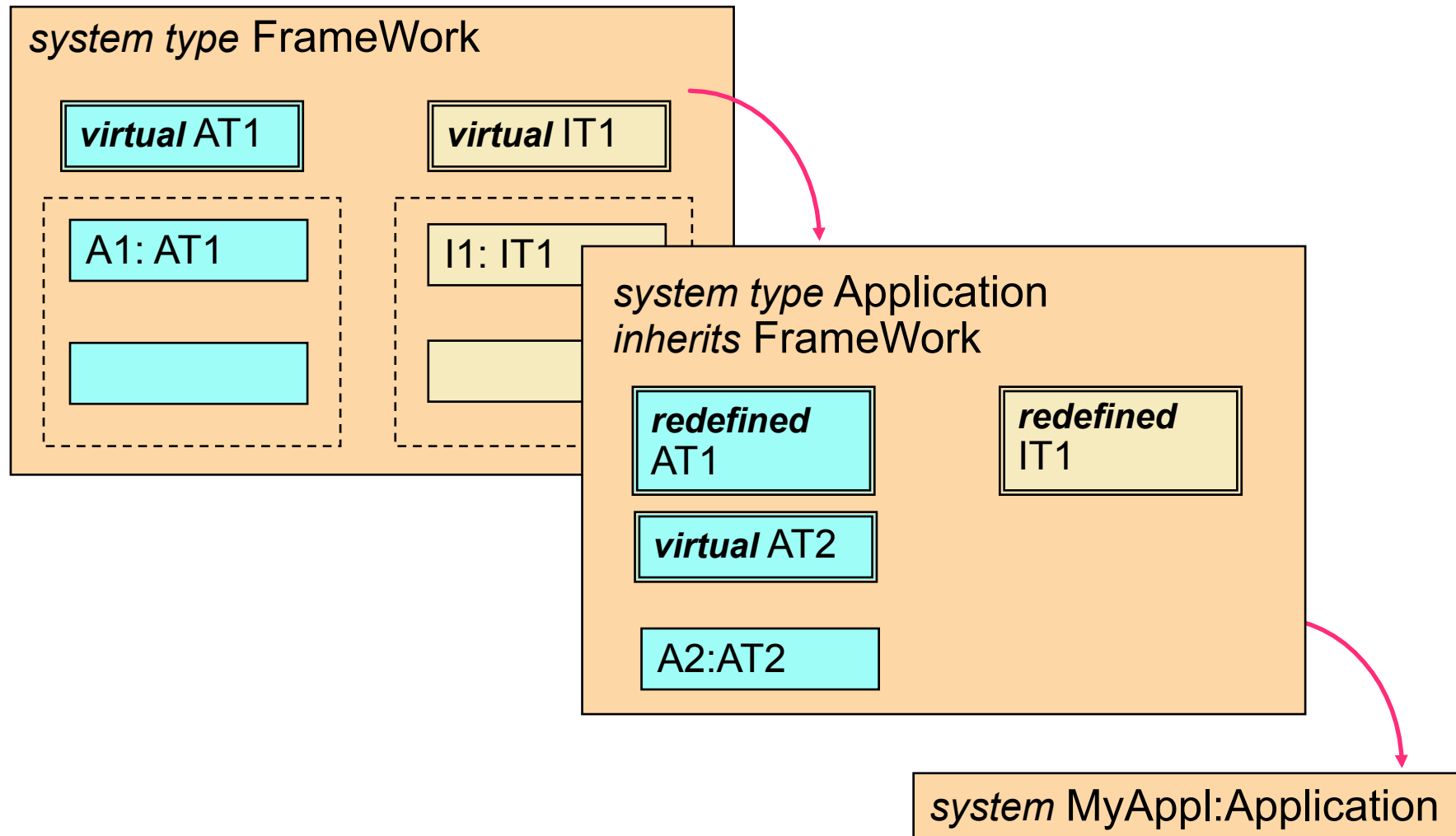
The complete SDL description as a Framework



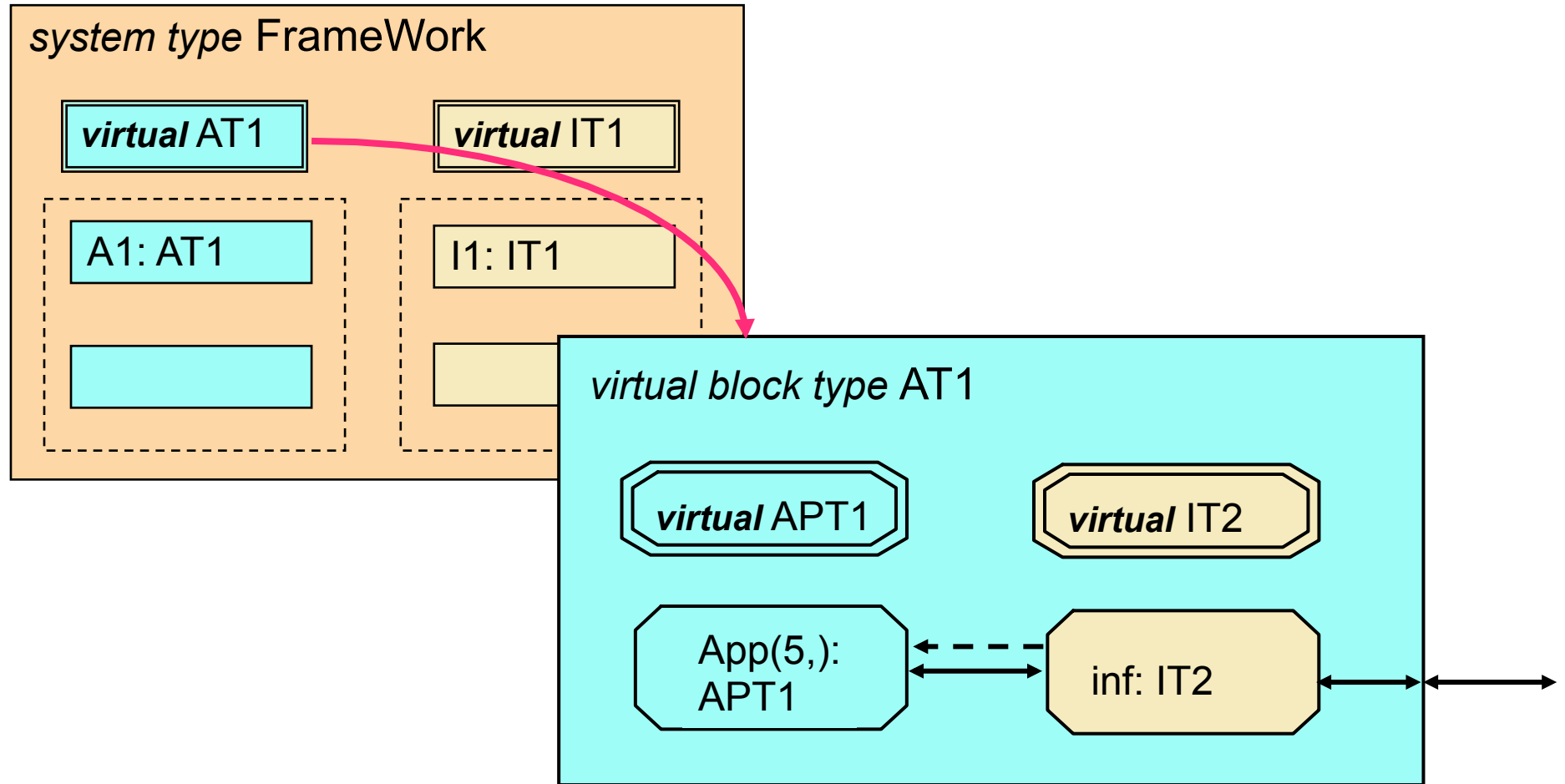
Benefits:

- Separation for development, maintenance and portability
- Combination for code generation and system handling

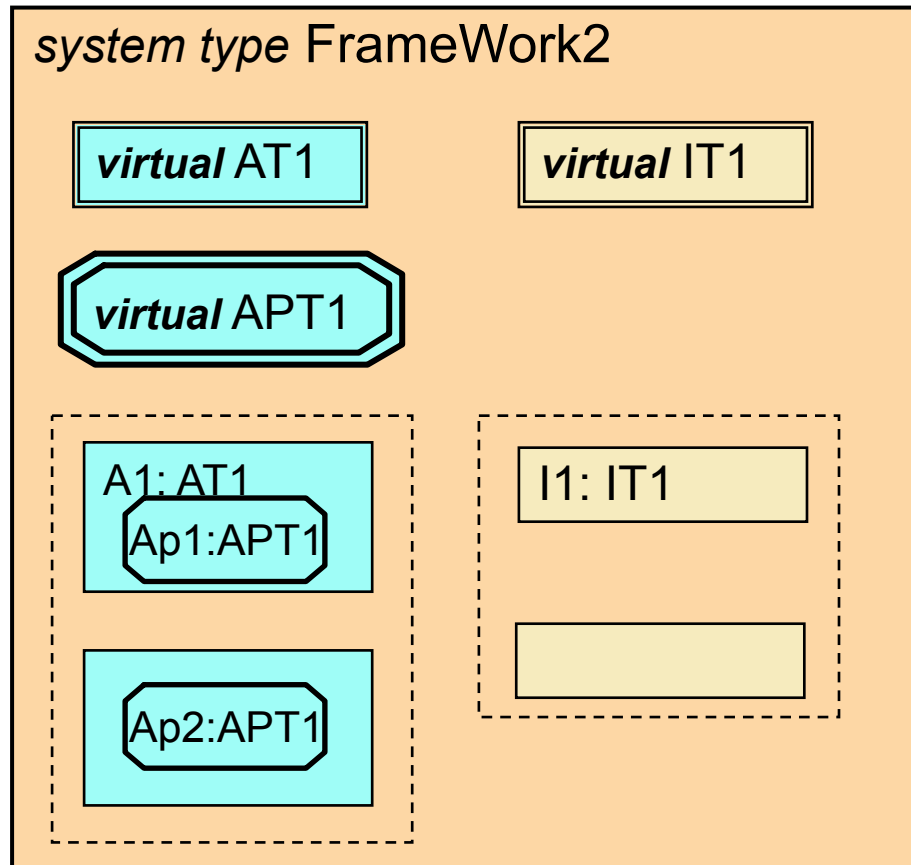
A Framework as a System Type



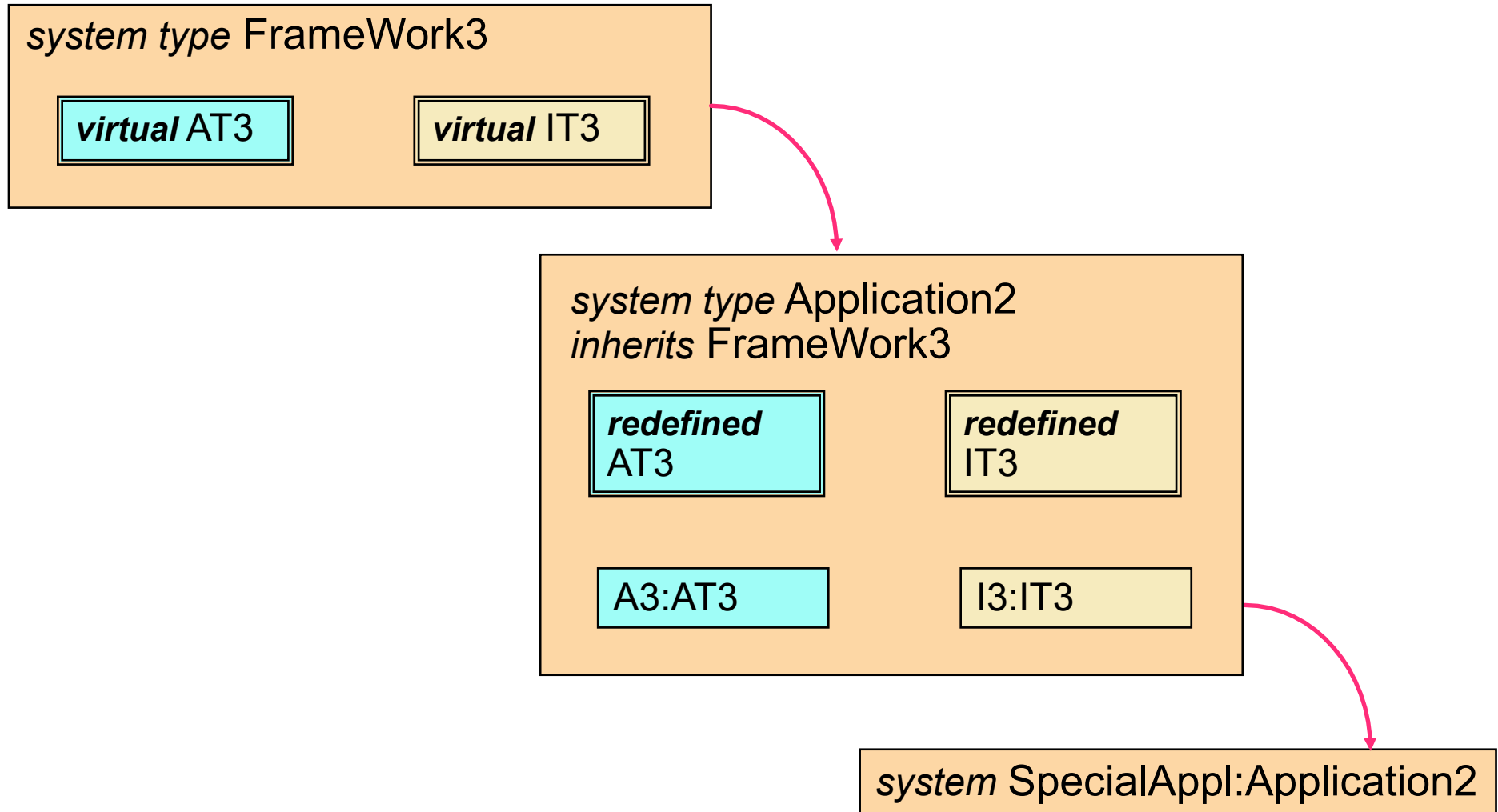
An Application Block



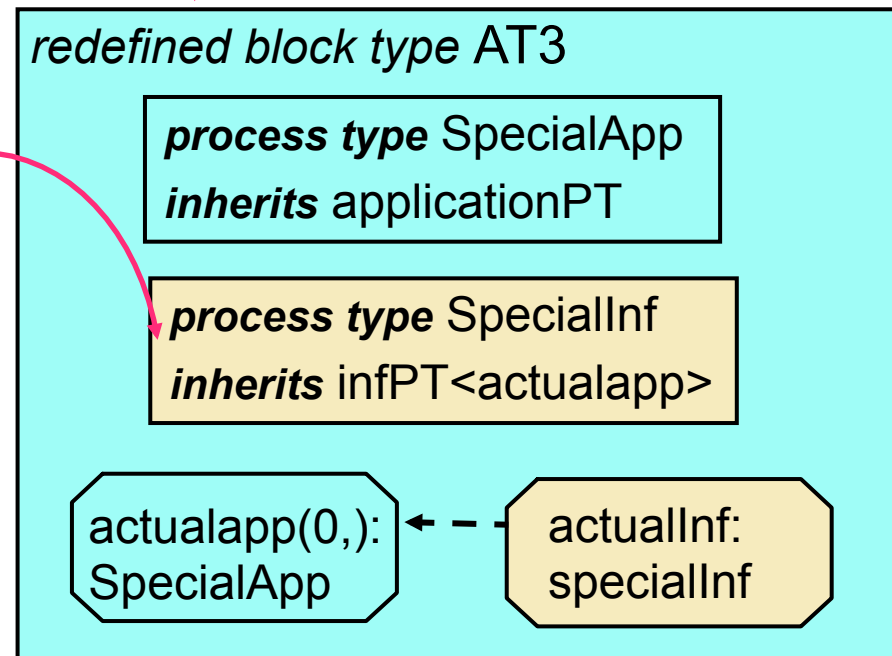
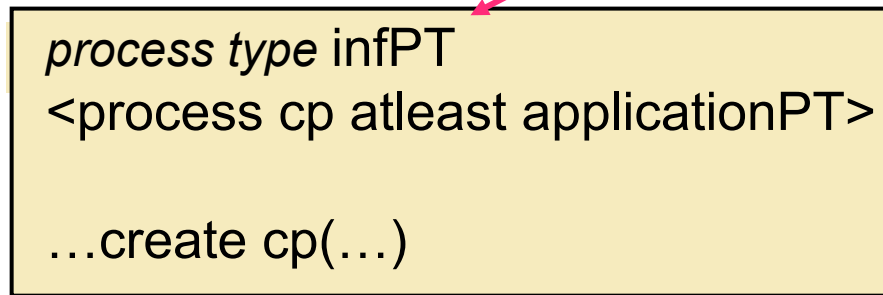
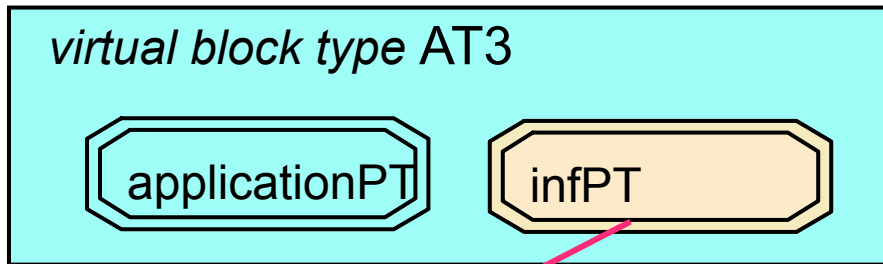
If an Application Process is common to many parts:



A Framework with no instances

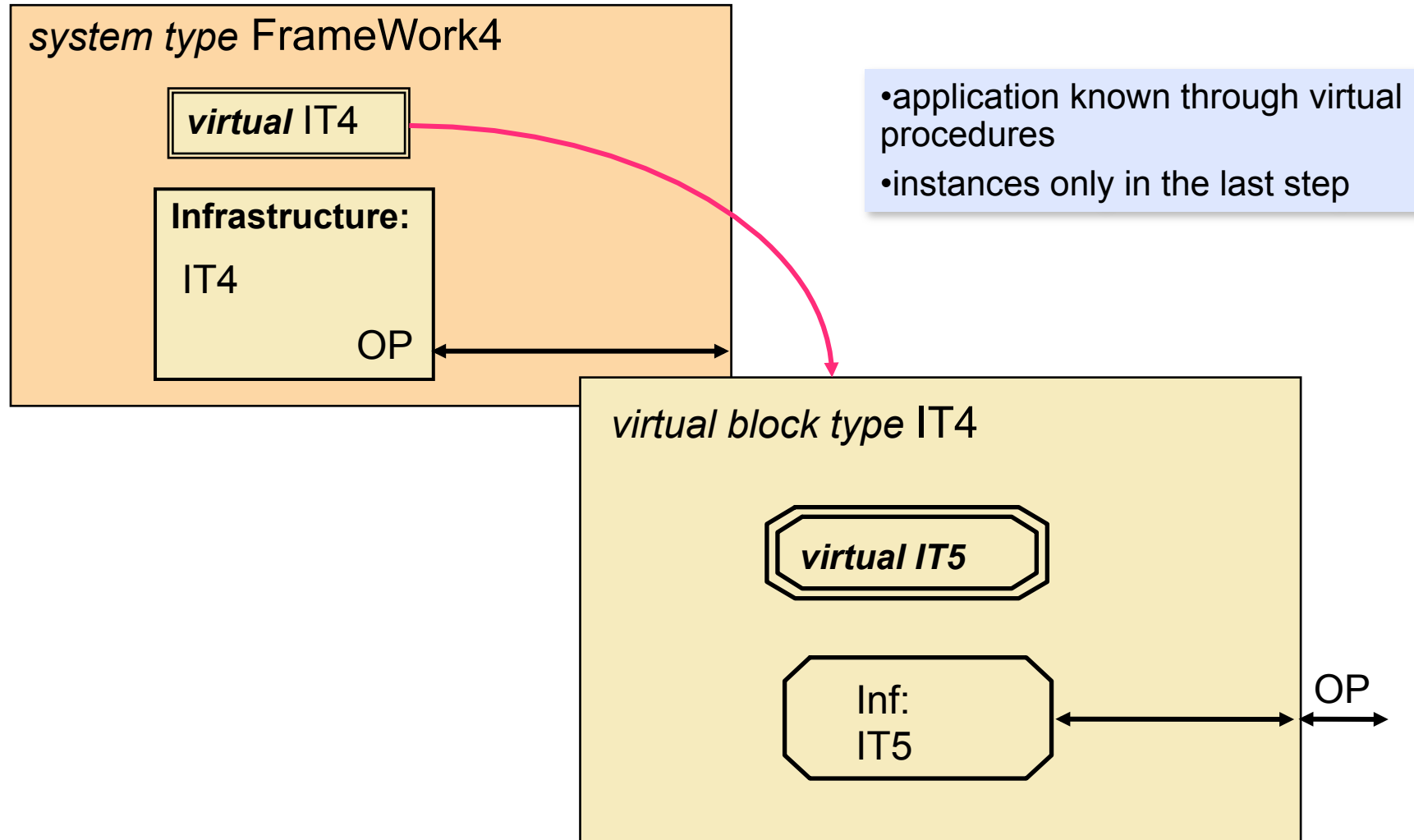


A virtual block type without instances

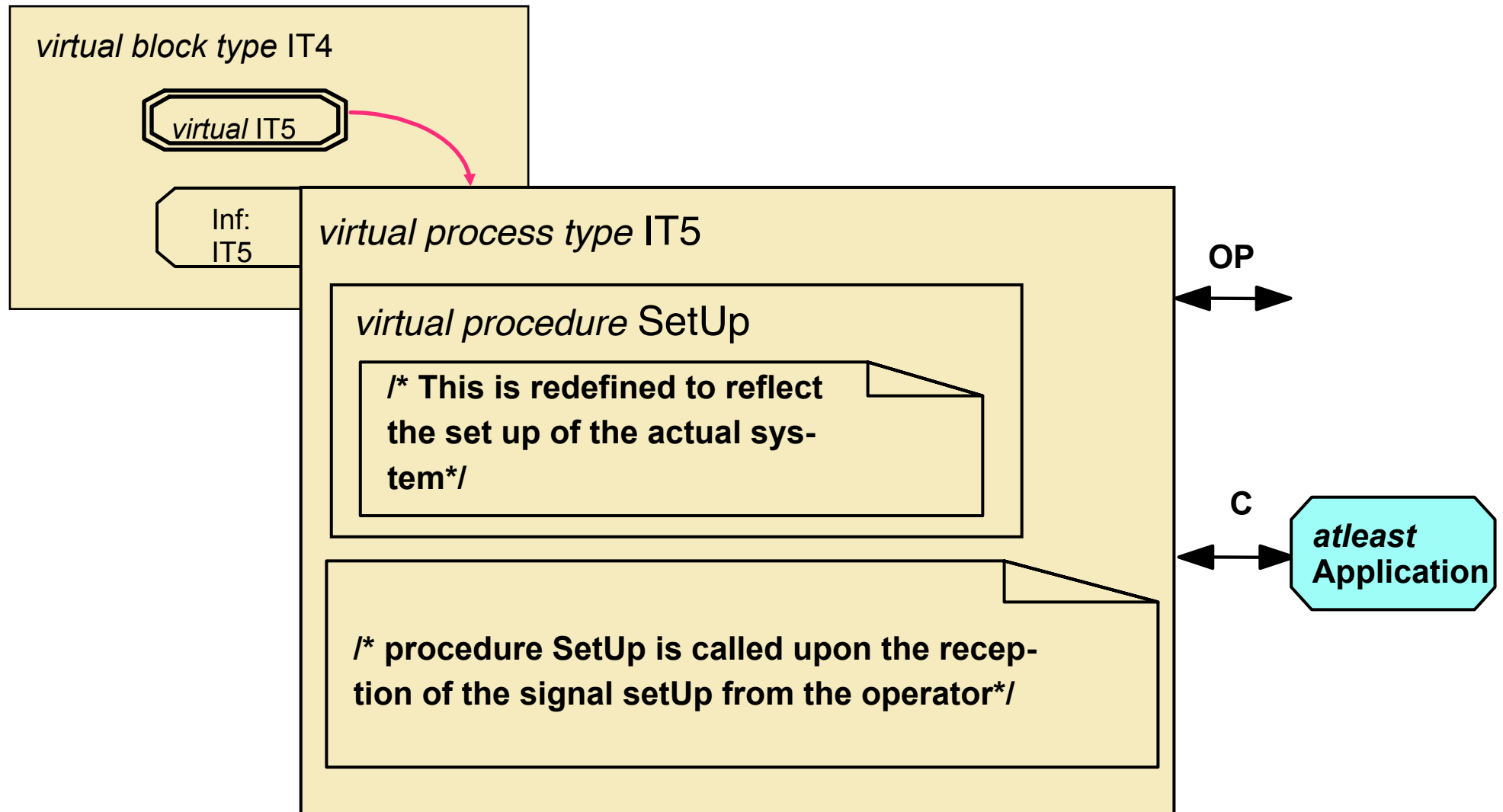


- application known through context parameters
- instances only in the last step

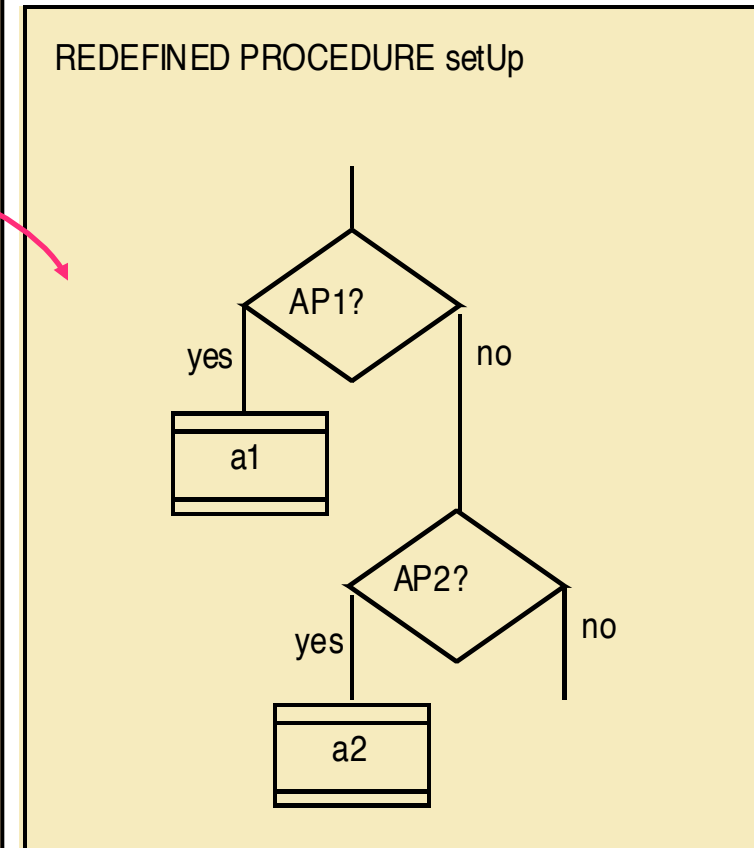
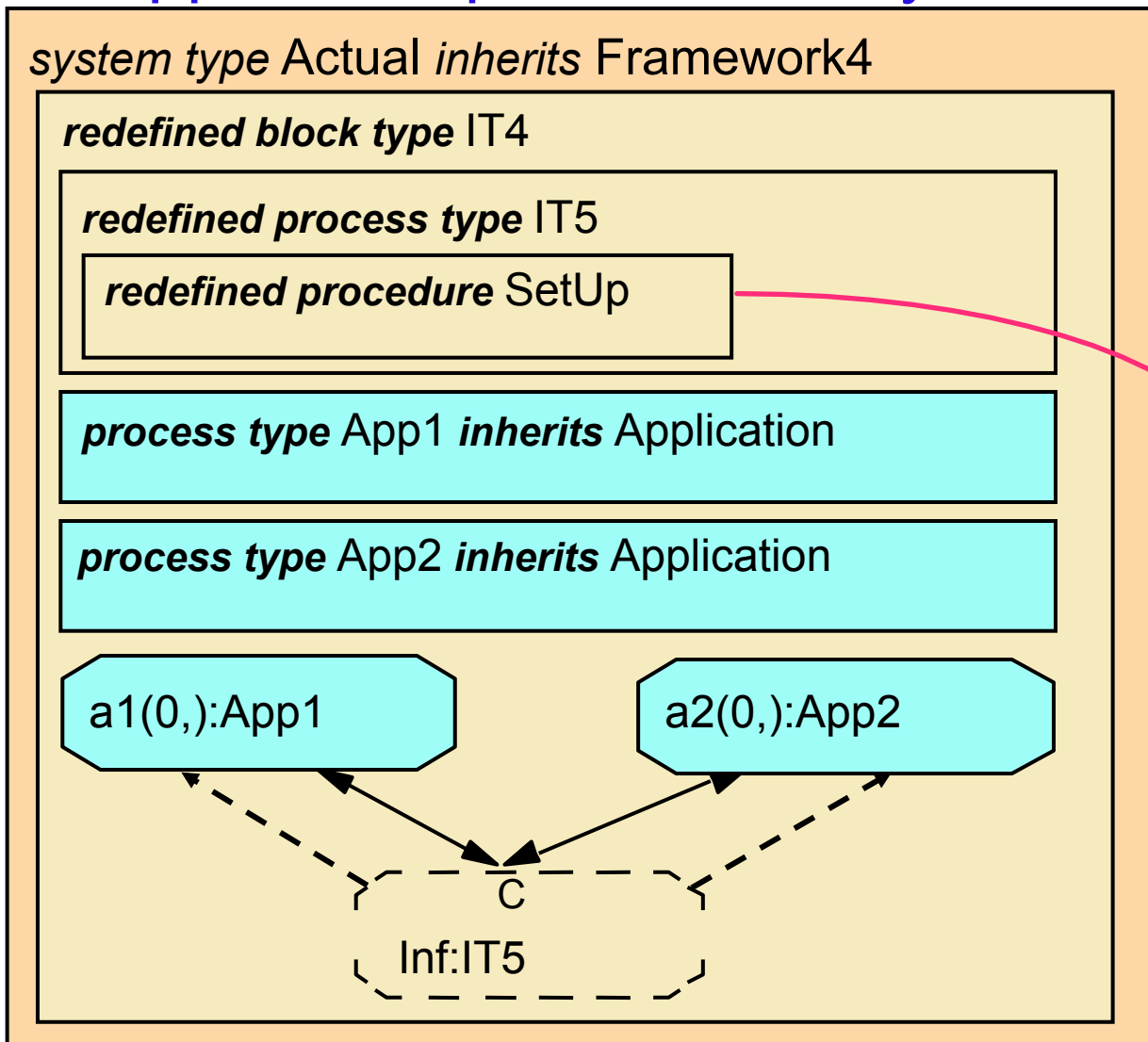
A Framework with virtual creation procedures



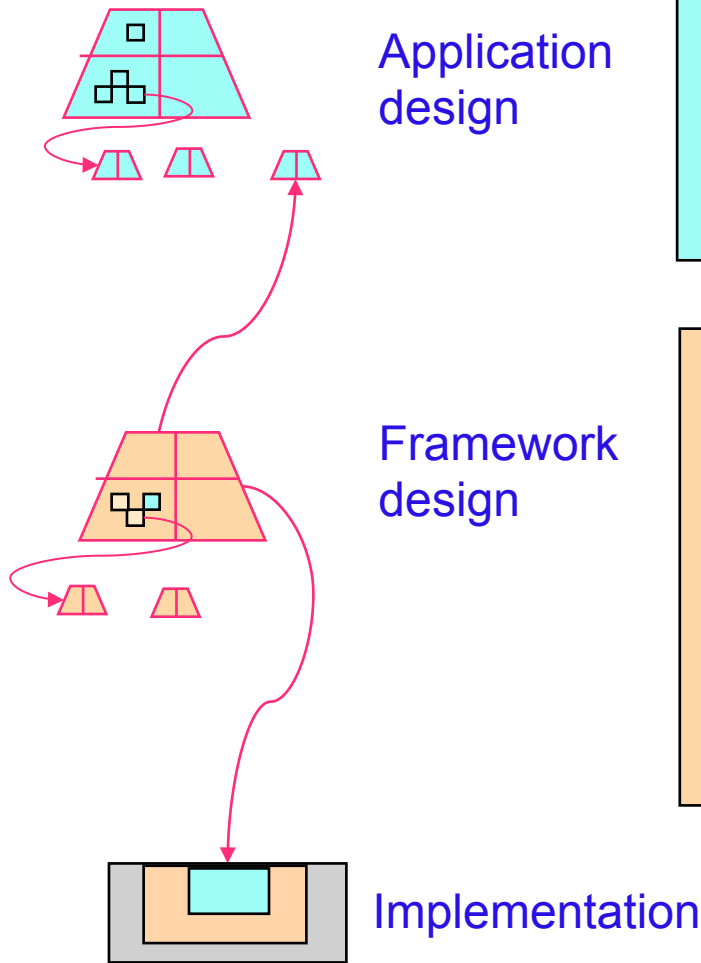
SetUp as a virtual procedure



Application part created by virtual procedure



Using Framework in AXE-10



Application design

Framework design

Implementation

block type SubServices

call(0,):
callhandler

sub(0,):
subhandler

block type SS inherits AXE10Block

redefined
blockproc

bp:
blockproc

call(0,):
callhandler

sub(0,):
subhandler

Focus on service behaviour
Simulation, validation of services

Focus on platform and implementation issues
Simulation of start/restart, size alteration, ...
Code generation

PLEX, C

Framework oriented development

- Support SDL development
 - *on a platform of pre-defined set of types that capture the implementation specific properties,*
 - *by specialising these types to the needs of the actual applications.*
- Support the separation of *application-* and *implementation* specific parts of applications
- Support the definition of architectures of combined application- and implementation specific parts defined in such a way that the application specific parts can be modified without modifying the architecture.