

The diagram illustrates the architecture of a smart language. It is organized into layers from top to bottom: 1. Processor / Platform (yellow oval). 2. Smart Language Specification (green rectangle). 3. Smart Interpreter (blue rectangle) and Smart Compiler (purple rectangle). 4. Target Platforms (divided into three columns by vertical grey lines): ARM (orange), x86/x64 (green), and wasm (pink). The ARM column has two sub-blocks: iOS/Mac and Android. The x86/x64 column has one sub-block: PC. The wasm column has one sub-block: WebBrowser. Horizontal dashed teal lines separate the top three layers, and the bottom layer is further divided by vertical grey lines.

Processor / Platform

Smart Language Specification

Smart Interpreter

Smart Compiler

ARM

x86/x64

wasm

iOS/Mac

Android

PC

WebBrowser

Compiler/Interpreter Steps

Parse Source Code To Abstract Node Tree

Initialize Built-In Types

Evaluate Types from Nodes and Validate Types

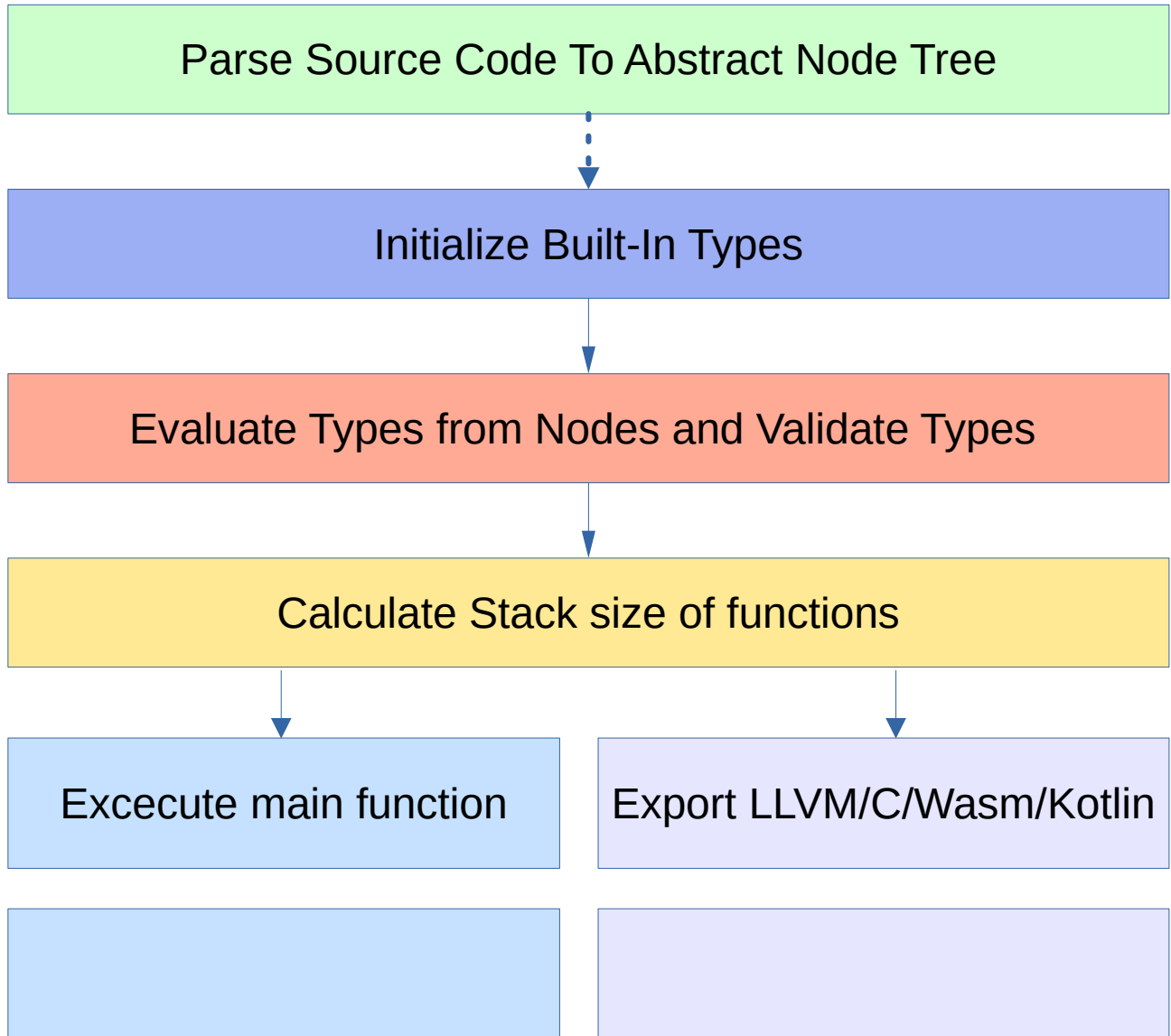
Calculate Stack size of functions

Excecute main function

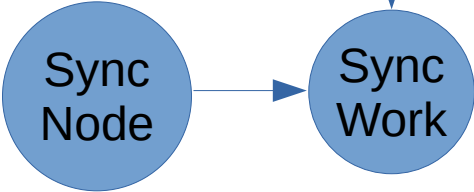
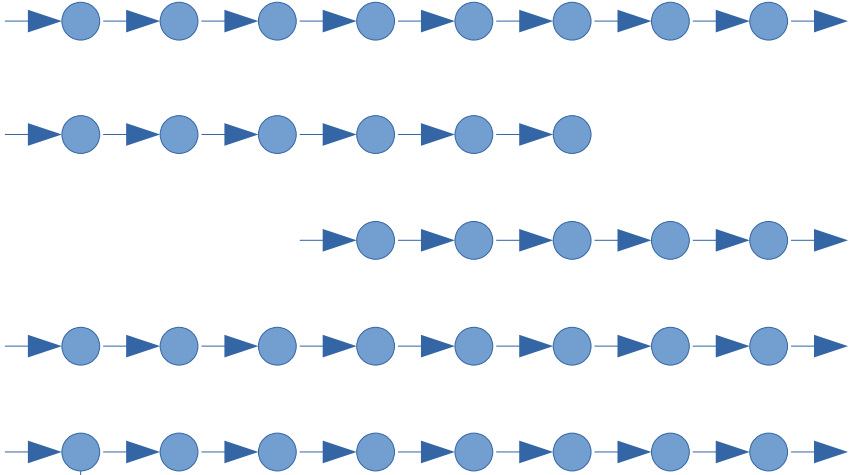
Export LLVM/C/Wasm/Kotlin

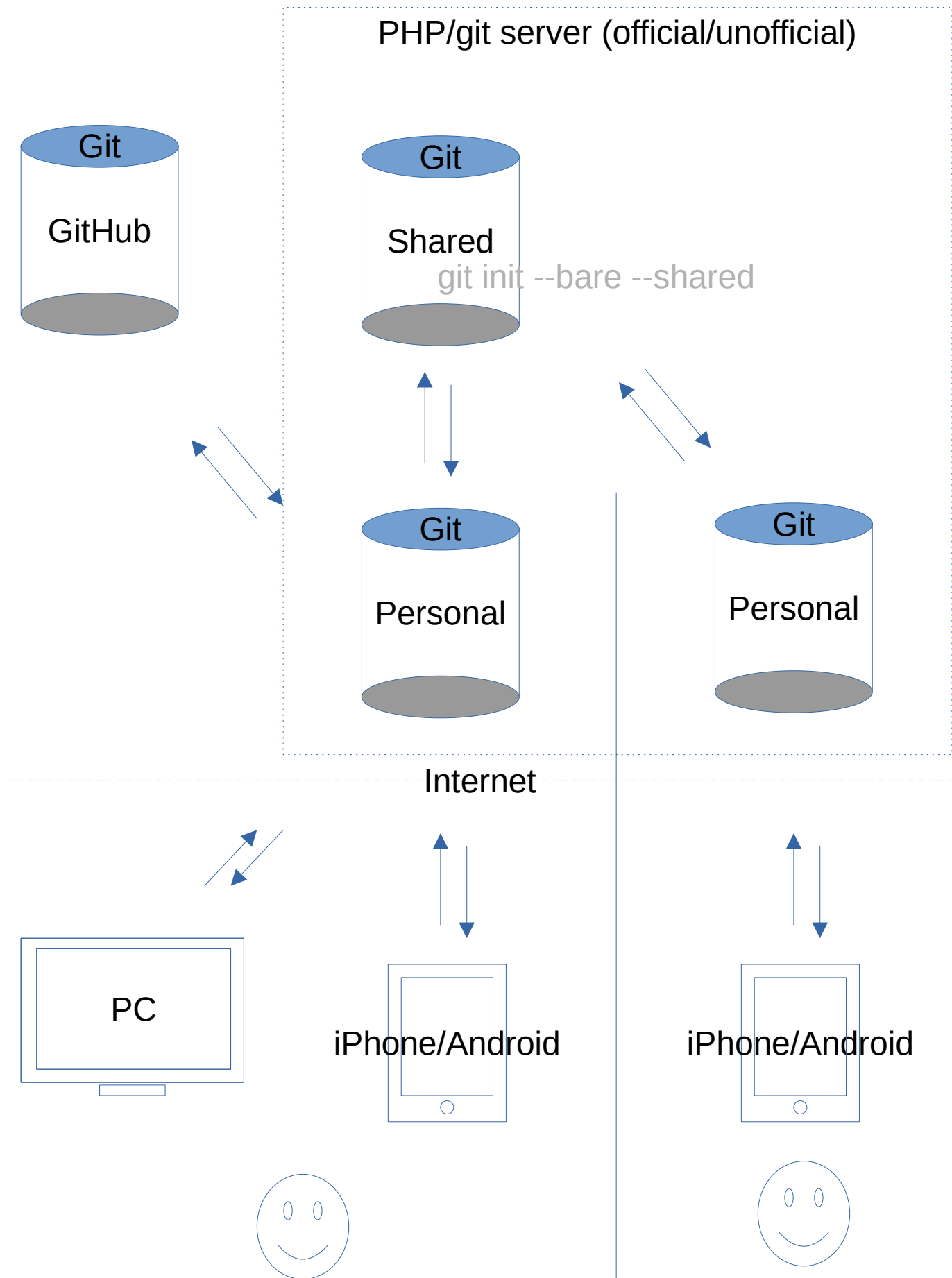
Interpreter

Compiler

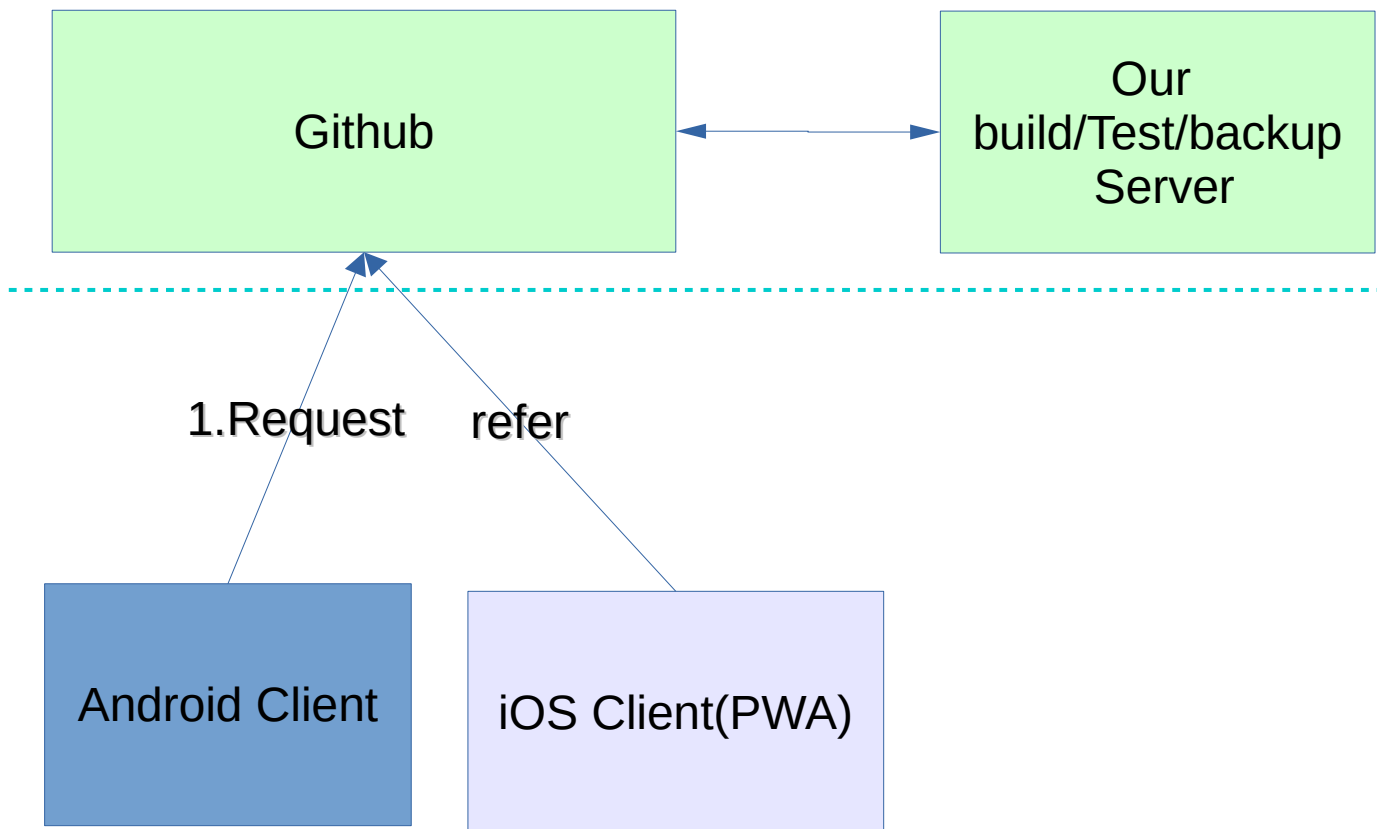


Central Work Queues





Online Build System



Wasm multiple modules

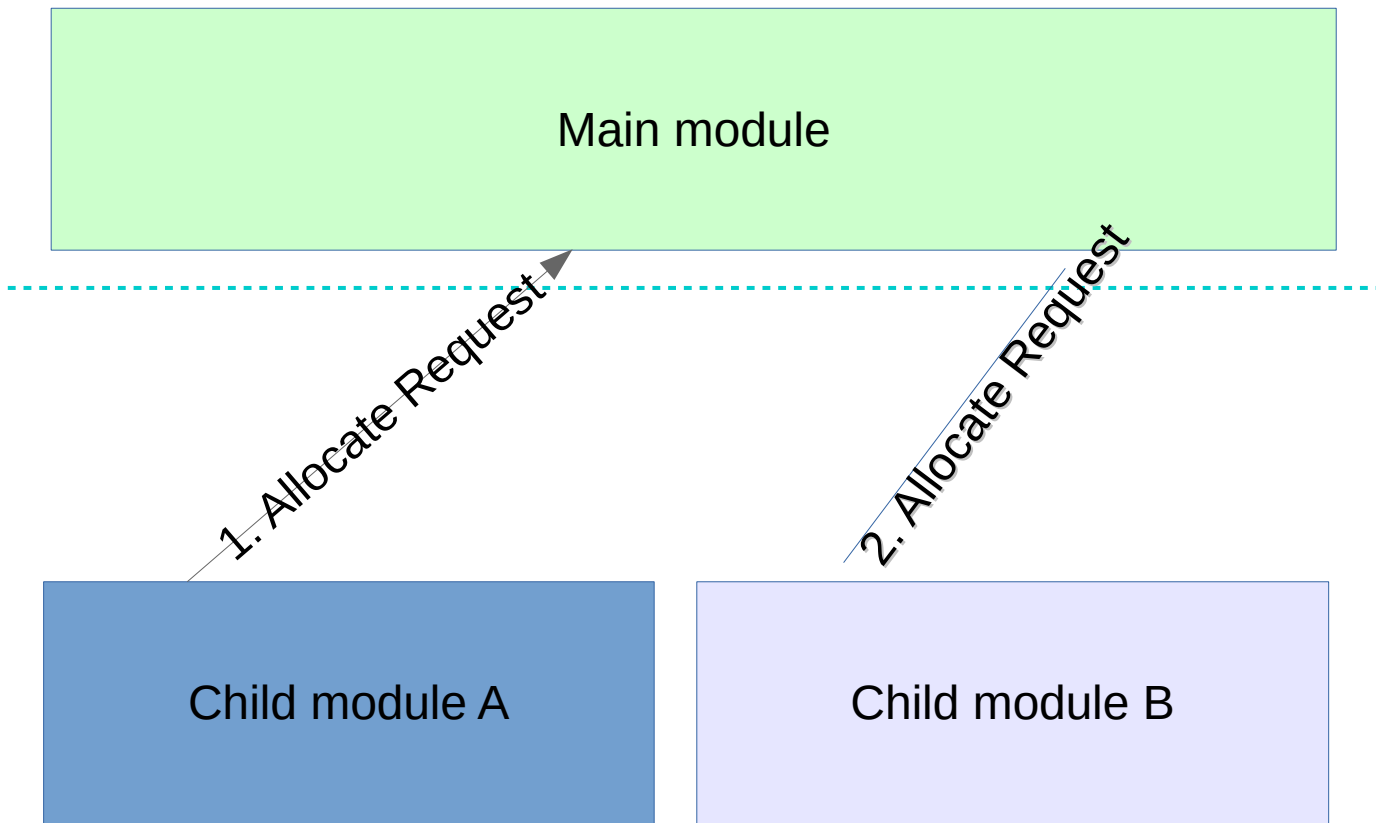
Main module

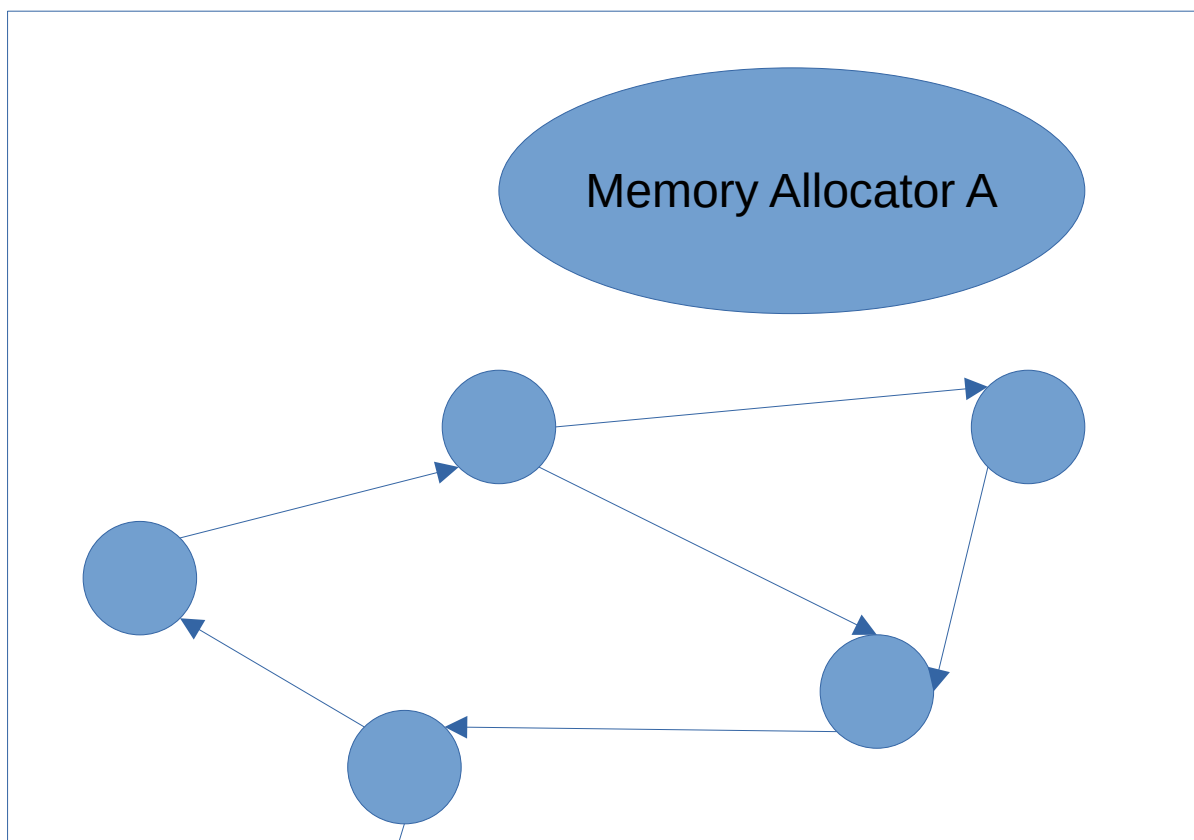
1. Allocate Request

2. Allocate Request

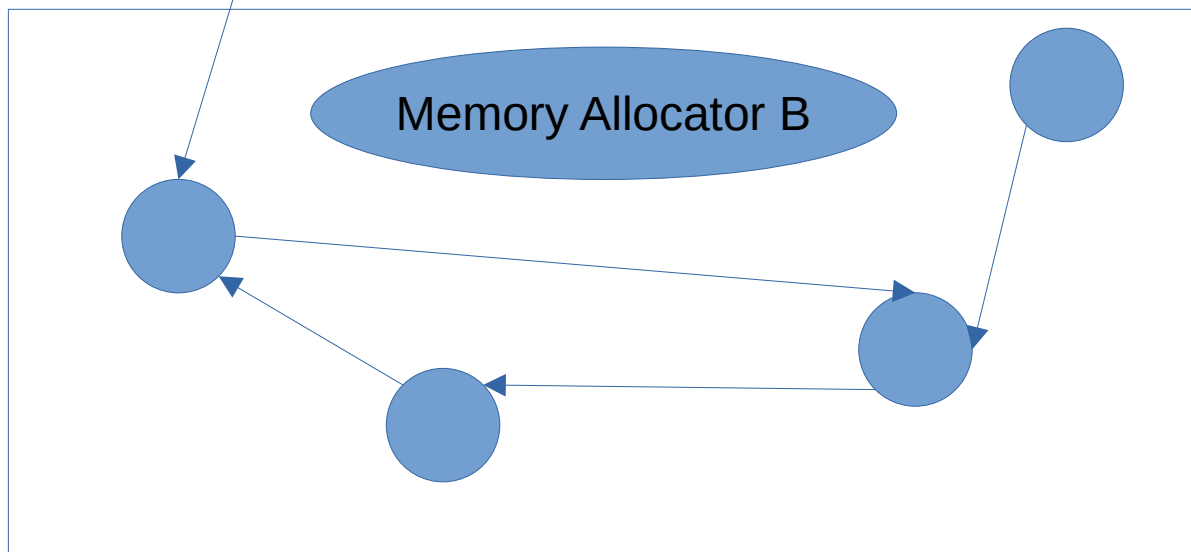
Child module A

Child module B





Not allowed



Lock A

Root

Context

A

Lock B

B

