DESIGN ENVIRONMENT FOR FPGA-BASED DESIGNS

BOTTOM-LEVEL DESIGN

OUTLINE

- Introduction
- Top-Level code generator
- Bitstream generator
- CPU communication module
- Summary

OUTLINE

- ➤ Introduction
- ➤ Top-Level code generator
- ➤ Bitstream generator
- ➤ CPU communication module
- ➤ Summary

INTRODUCTION

- Main purpose:
  - Develop a software environment allowing user to accomplish their design on it.

INTRODUCTION

- Main purpose:
  - Develop a software environment allowing user to accomplish their design on it.
- System specification

INTRODUCTION

- Main purpose:
  - Develop a software environment allowing user to accomplish their design on it.
- System specification
  - A user interface

INTRODUCTION

- Main purpose:
  - Develop a software environment allowing user to accomplish their design on it.
- System specification
  - A user interface
  - Conclude user design into codes

INTRODUCTION

- Main purpose:
  - Develop a software environment allowing user to accomplish their design on it.
- System specification
  - A user interface
  - Conclude user design into codes
  - Program codes into device

INTRODUCTION

- Main purpose:
  - Develop a software environment allowing user to accomplish their design on it.
- System specification
  - A user interface
  - Conclude user design into codes
  - Program codes into device
  - Manage data transmit during run-time

INTRODUCTION – MODULES

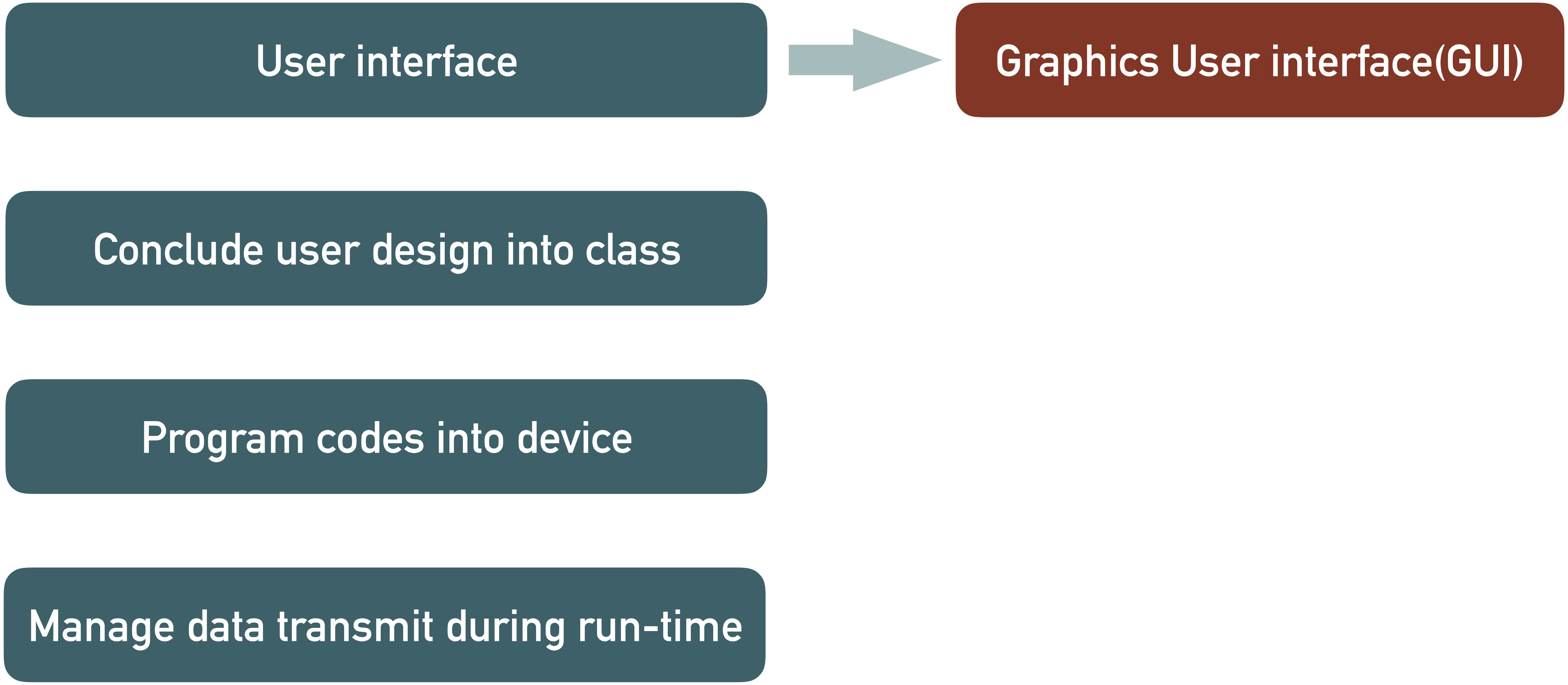
User interface

Conclude user design into class

Program codes into device

Manage data transmit during run-time

INTRODUCTION - MODULES



User interface

Graphics User interface(GUI)

Conclude user design into class

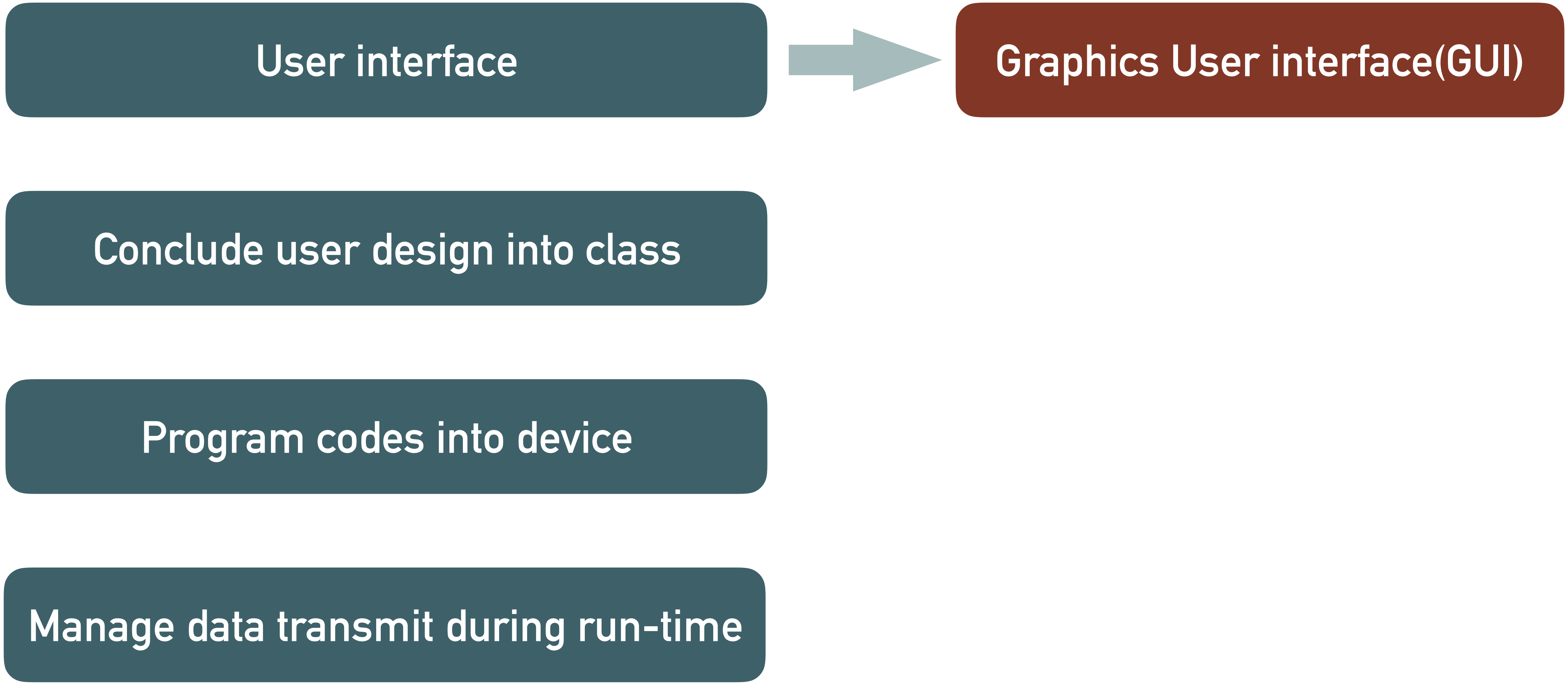
Program codes into device

Manage data transmit during run-time

INTRODUCTION - GRAPHICS USER INTERFACE (GUI)

- Graphics User Interface (GUI)
  - A graphics interface allowing user to build their customized system
  - User is able to embed IP cores into design
  - GUI collect system information for other modules

INTRODUCTION - MODULES



User interface

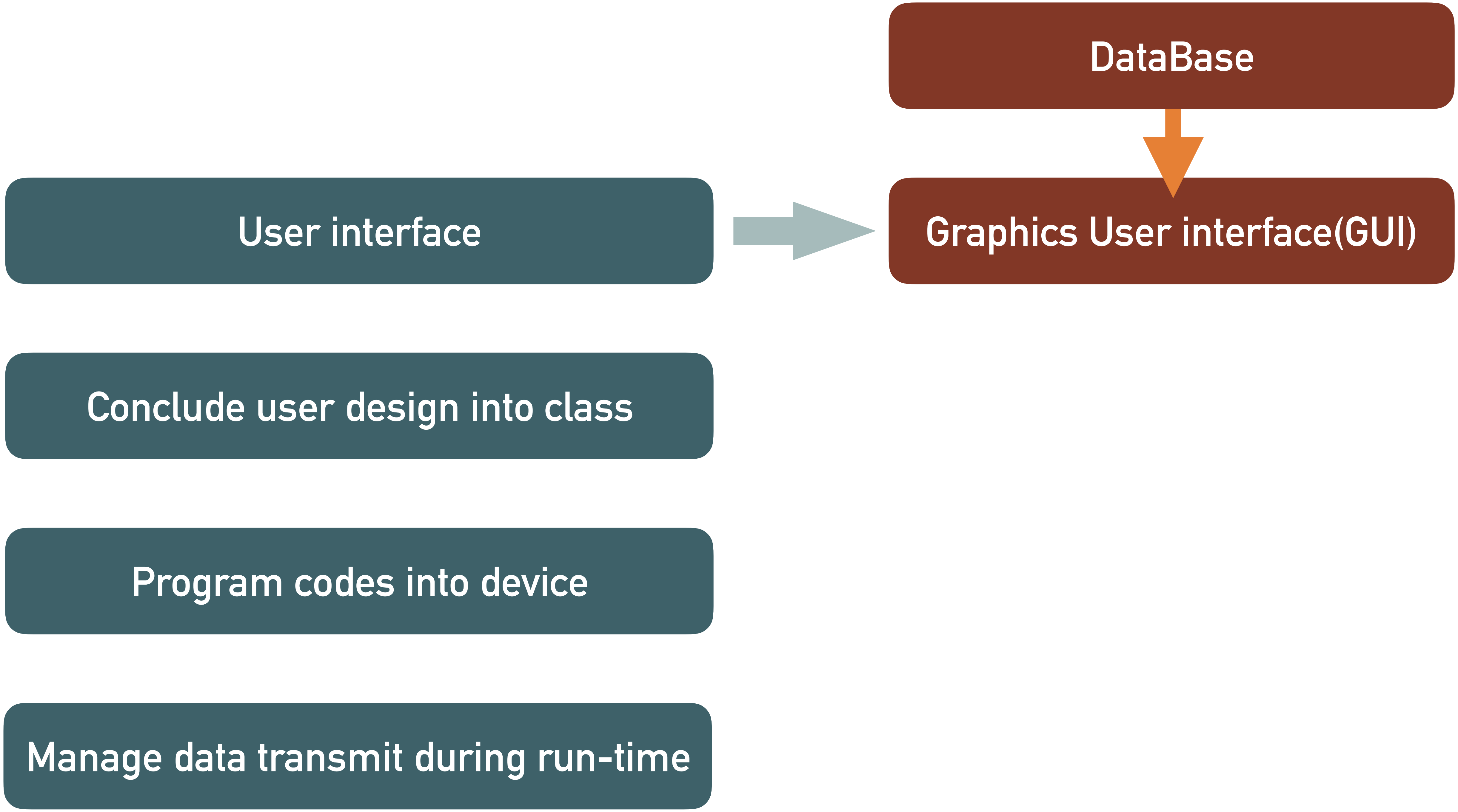
Graphics User interface(GUI)

Conclude user design into class

Program codes into device

Manage data transmit during run-time

INTRODUCTION - MODULES



DataBase

User interface

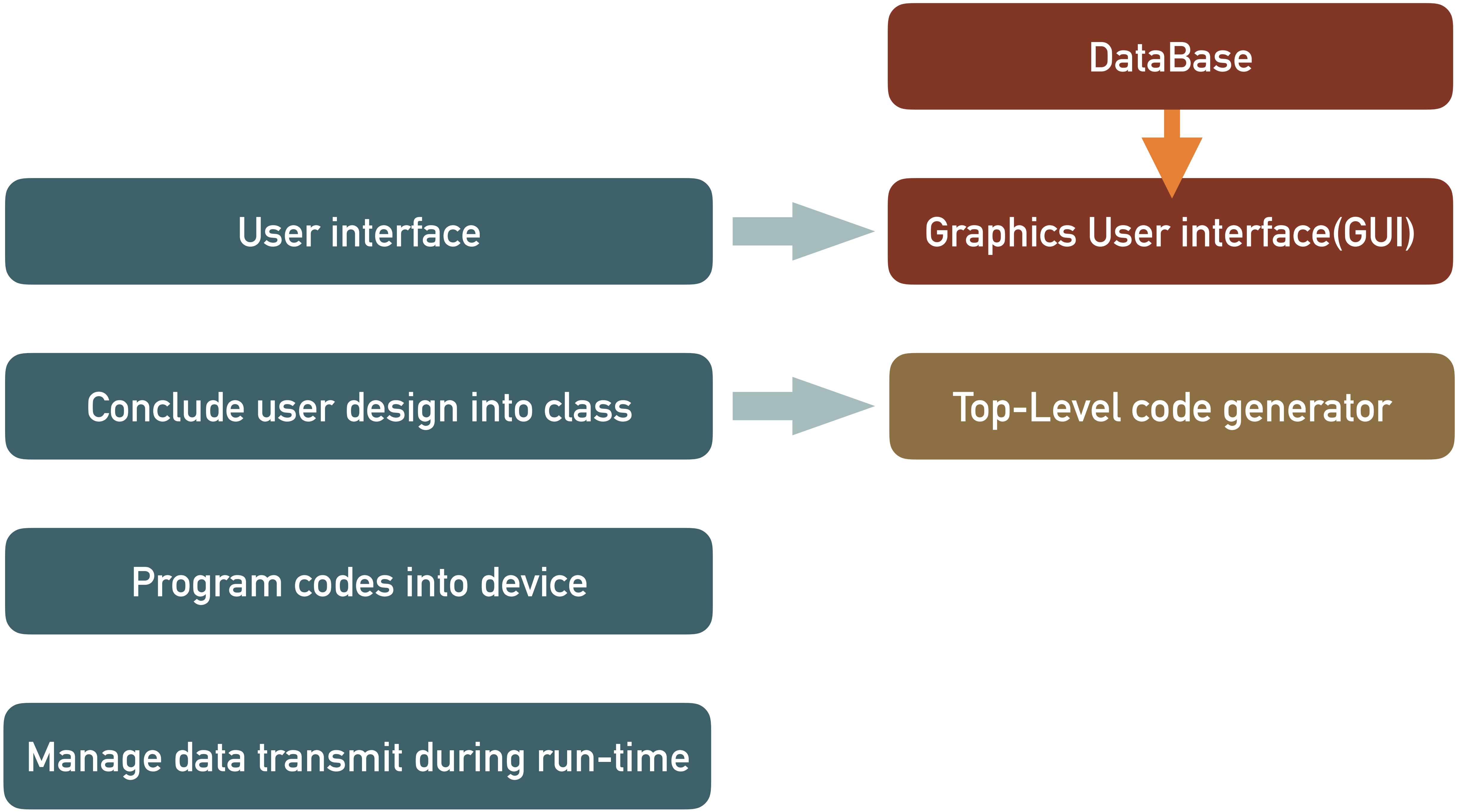
Graphics User interface(GUI)

Conclude user design into class

Program codes into device

Manage data transmit during run-time

INTRODUCTION – MODULES



DataBase

User interface

Graphics User interface(GUI)

Conclude user design into class

Top-Level code generator

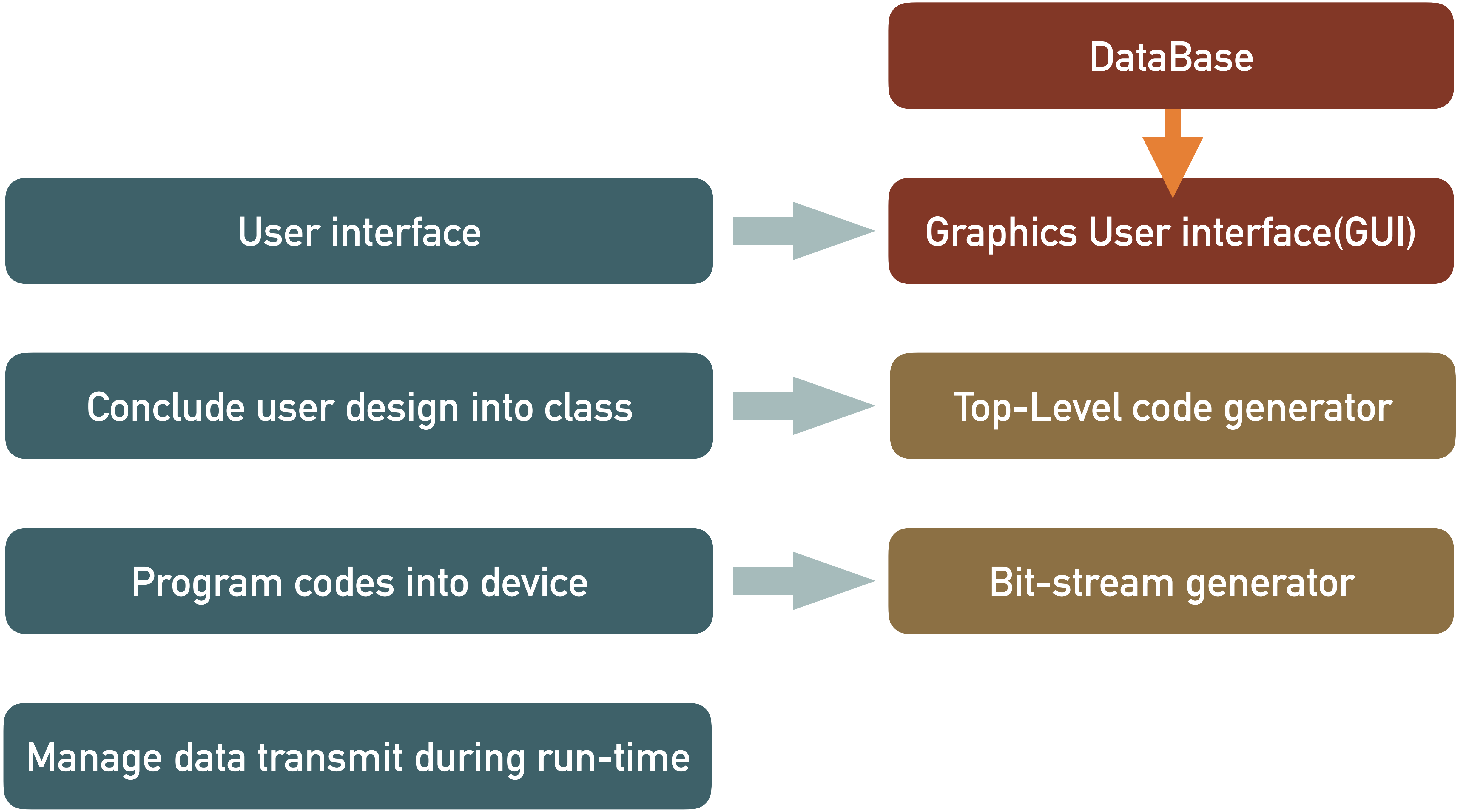
Program codes into device

Manage data transmit during run-time

INTRODUCTION - TOP-LEVEL CODE GENERATOR

- Top-level code generator
  - Take system information collected by GUI as input
  - Analyse the information
  - Generator top-level entity code of entire system in VHDL

INTRODUCTION – MODULES

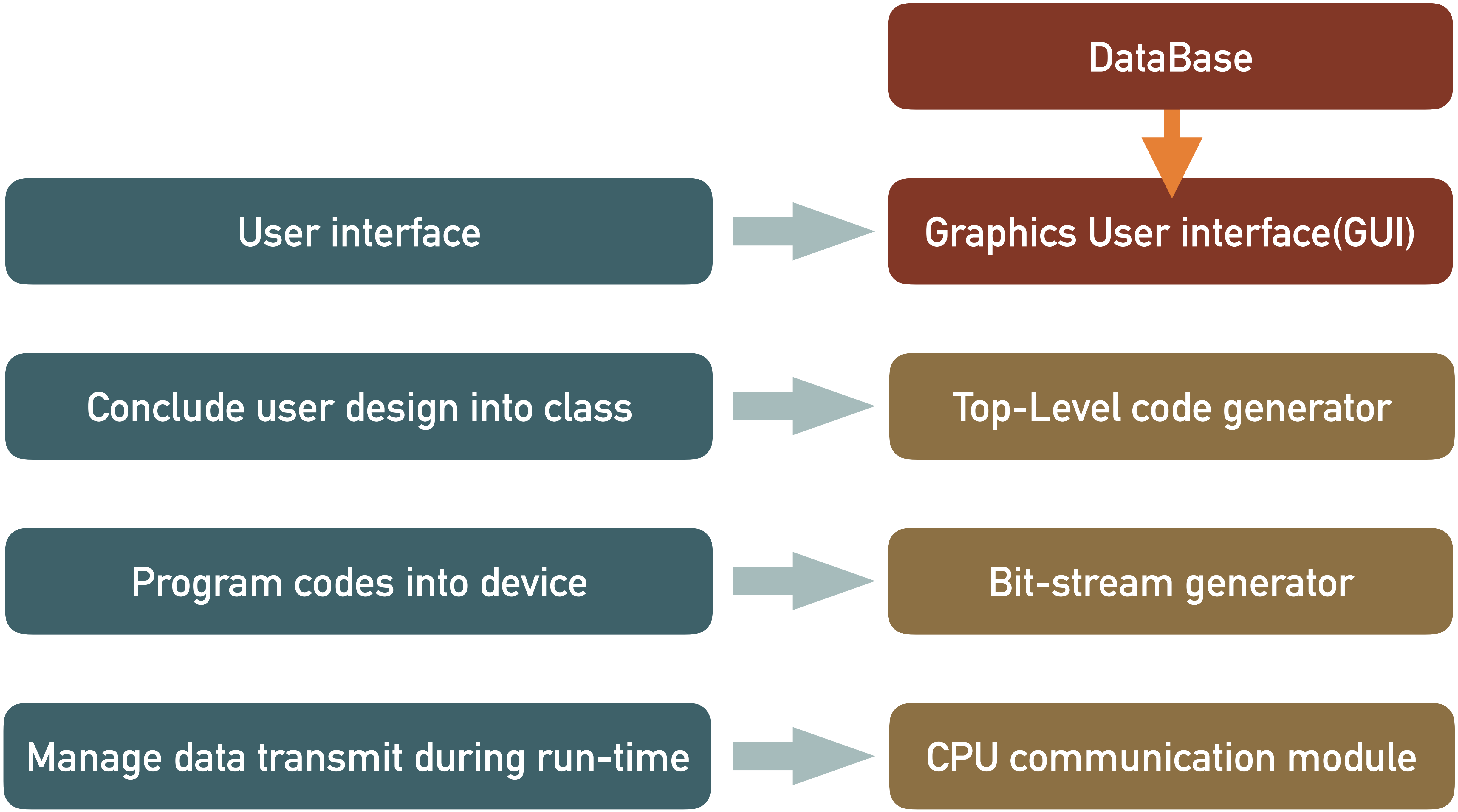


DataBase
User interface
Graphics User interface(GUI)
Conclude user design into class
Top-Level code generator
Program codes into device
Bit-stream generator
Manage data transmit during run-time

INTRODUCTION - BIT-STREAM GENERATOR

- Bit-stream generator
    - Based on software provided by Lattice which is “Diamond Lattice”
    - Take system information as input such as workspace path, Lattice directory.
    - Bit-stream generator generates the bit-stream used for programming FPGA

INTRODUCTION – MODULES



DataBase

User interface

Graphics User interface(GUI)

Conclude user design into class

Top-Level code generator

Program codes into device

Bit-stream generator

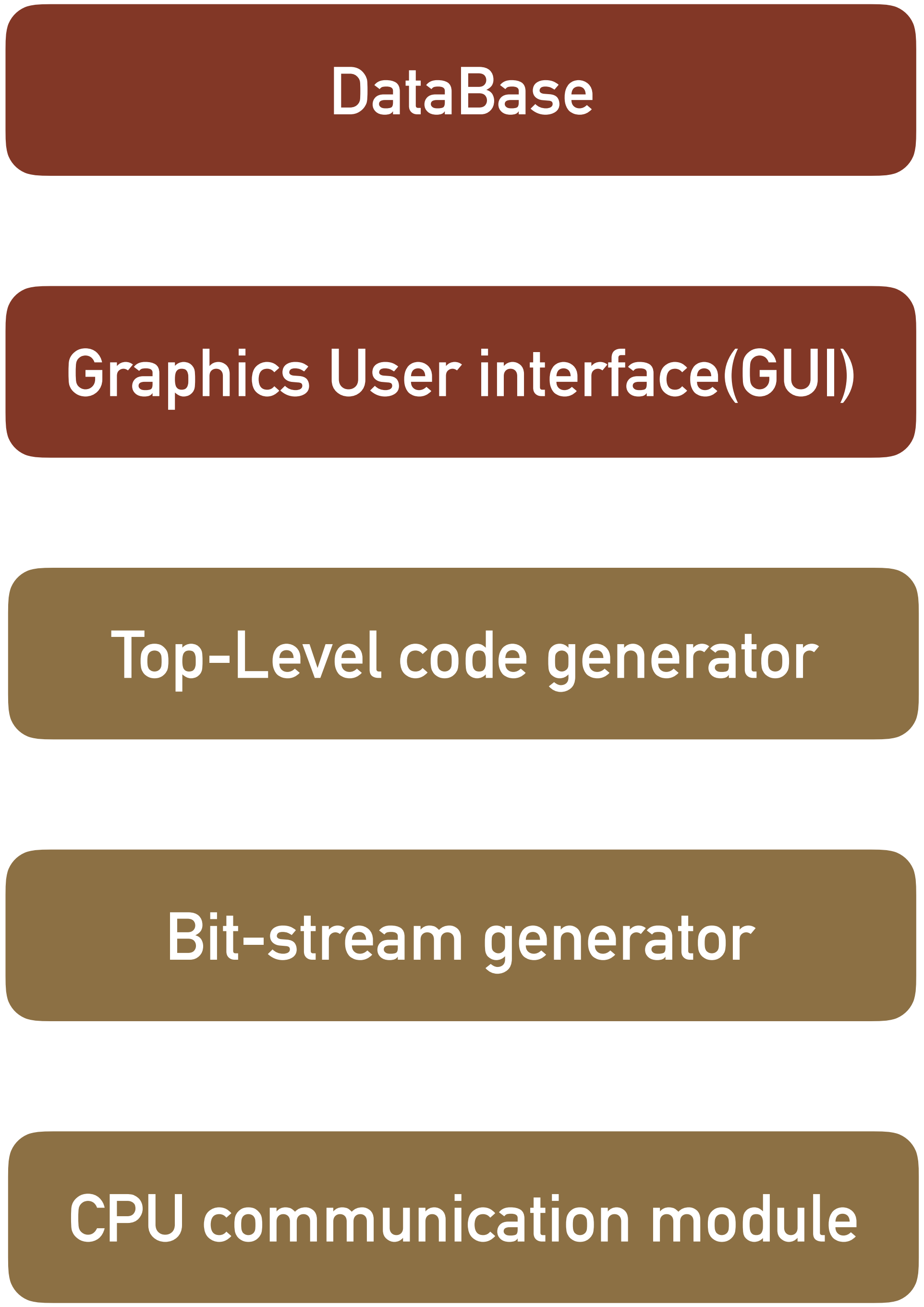
Manage data transmit during run-time

CPU communication module

INTRODUCTION - CPU COMMUNICATION MODULE

- CPU communication module
  - By previous modules, user is able to generate the bit-stream used for programming FPGA
  - This module is used to transmit data between CPU and FPGA in run-time

INTRODUCTION – WORKFLOW



DataBase

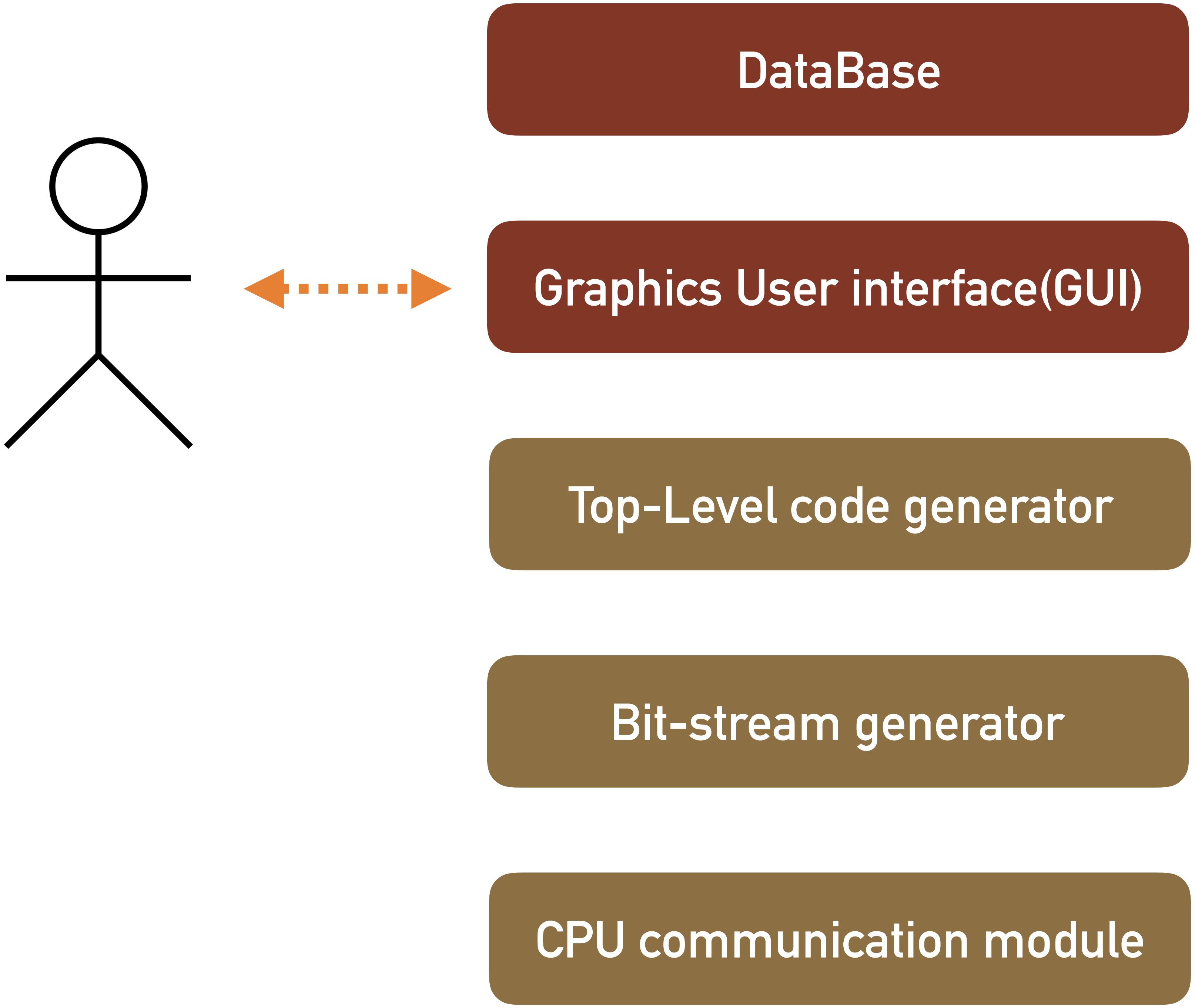
Graphics User interface(GUI)

Top-Level code generator

Bit-stream generator

CPU communication module

INTRODUCTION – WORKFLOW



DataBase

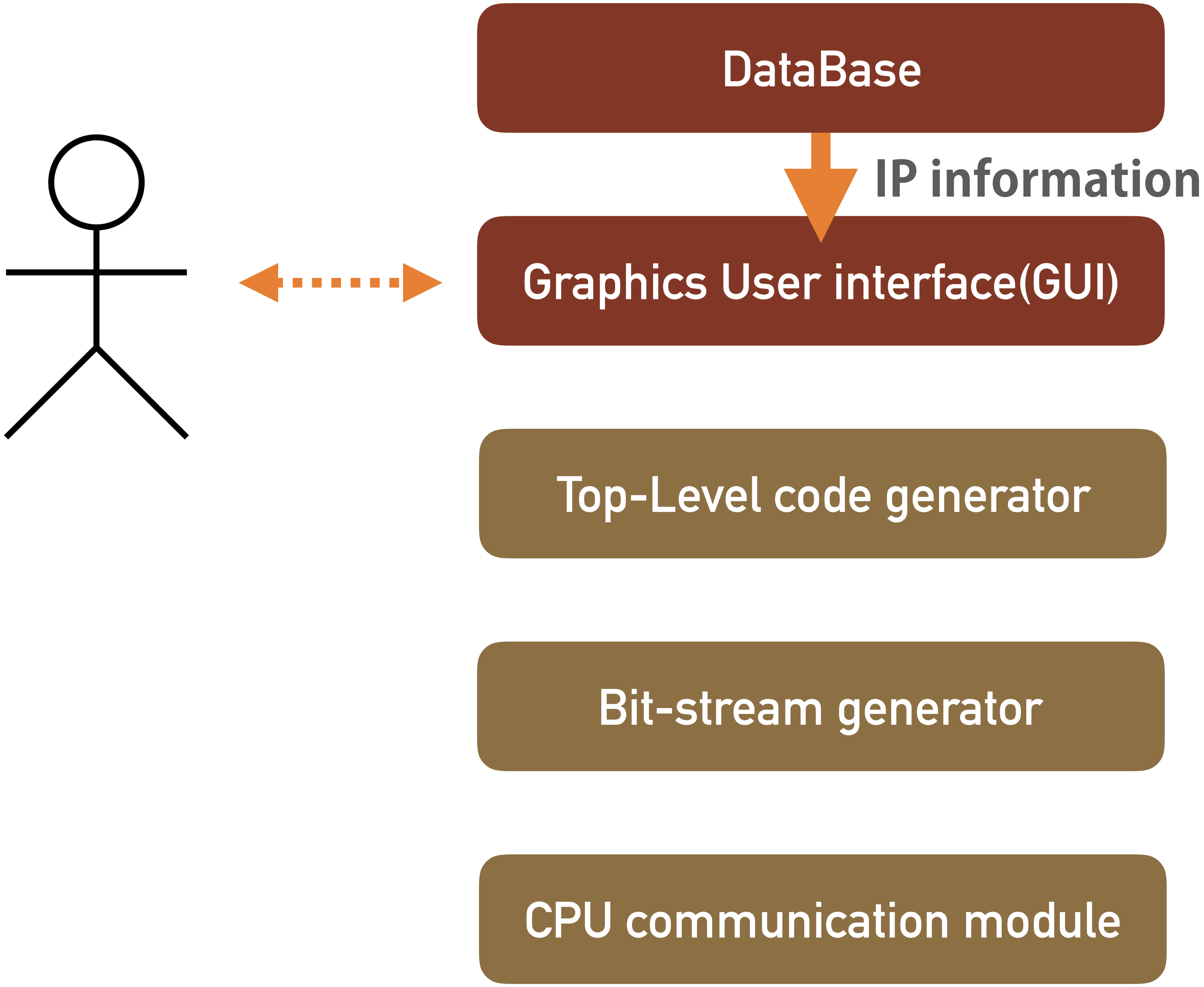
Graphics User interface(GUI)

Top-Level code generator

Bit-stream generator

CPU communication module

INTRODUCTION – WORKFLOW



DataBase

IP information

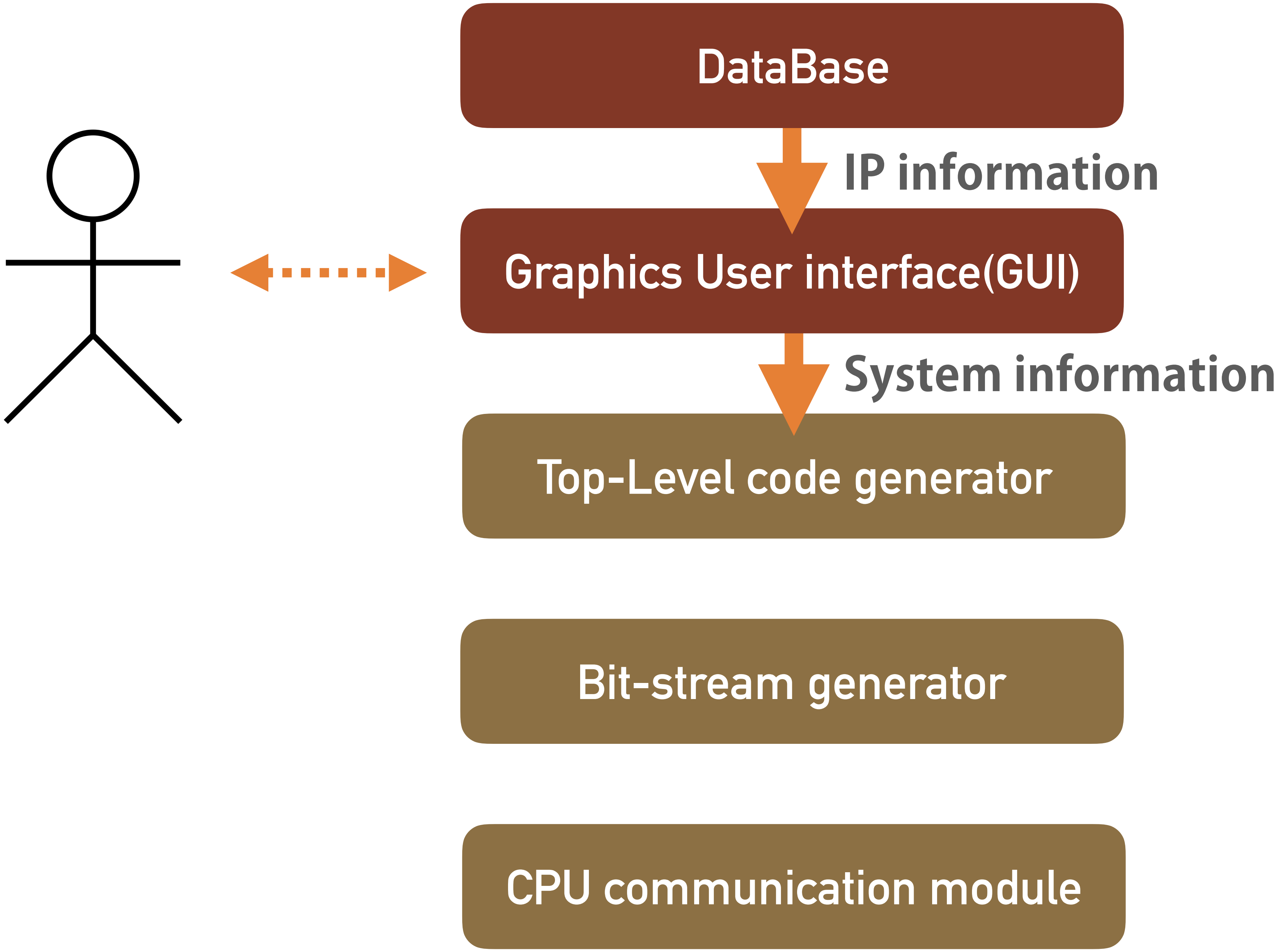
Graphics User interface(GUI)

Top-Level code generator

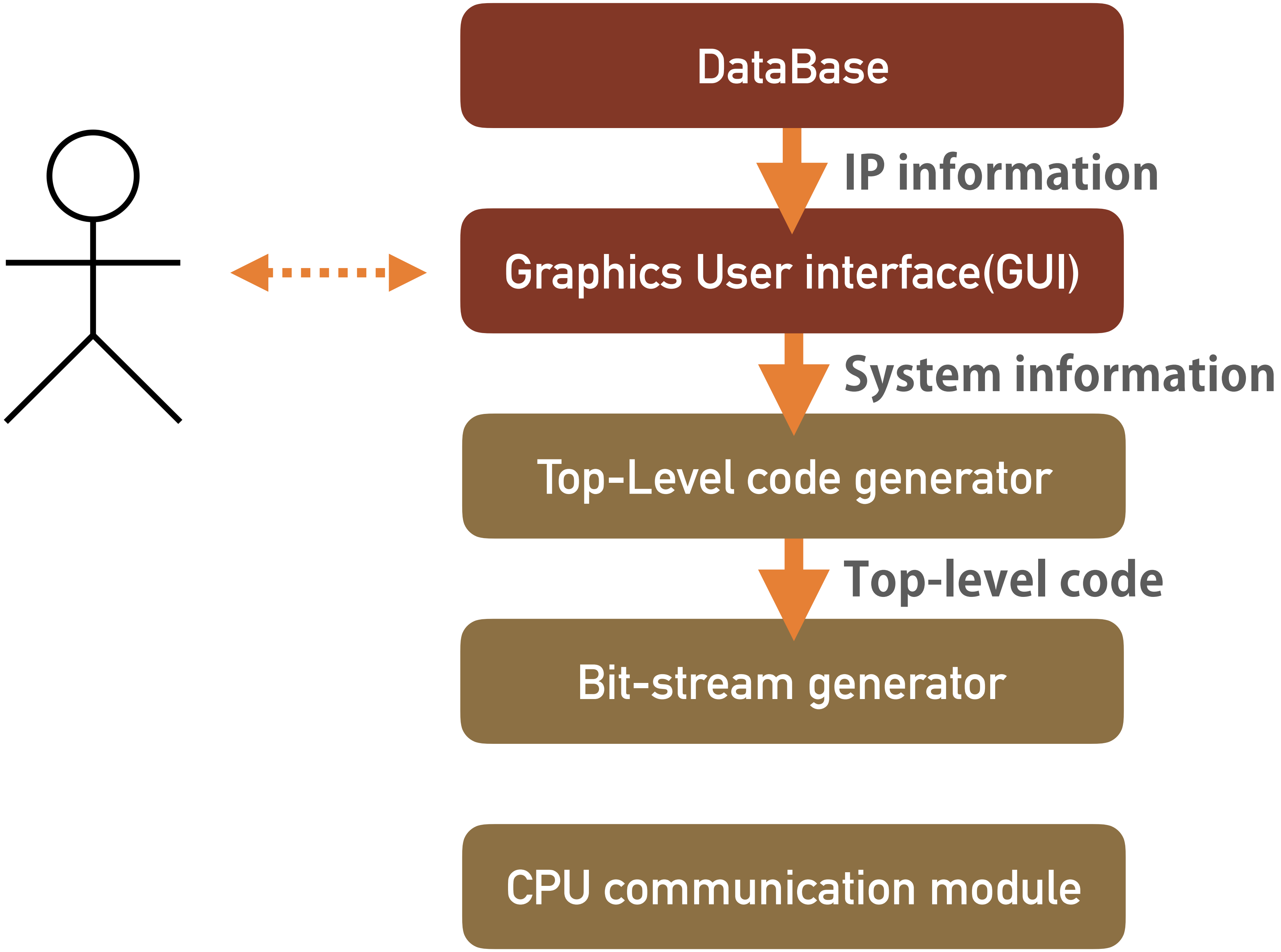
Bit-stream generator

CPU communication module

INTRODUCTION - WORKFLOW



INTRODUCTION – WORKFLOW



DataBase

IP information

Graphics User interface(GUI)

System information

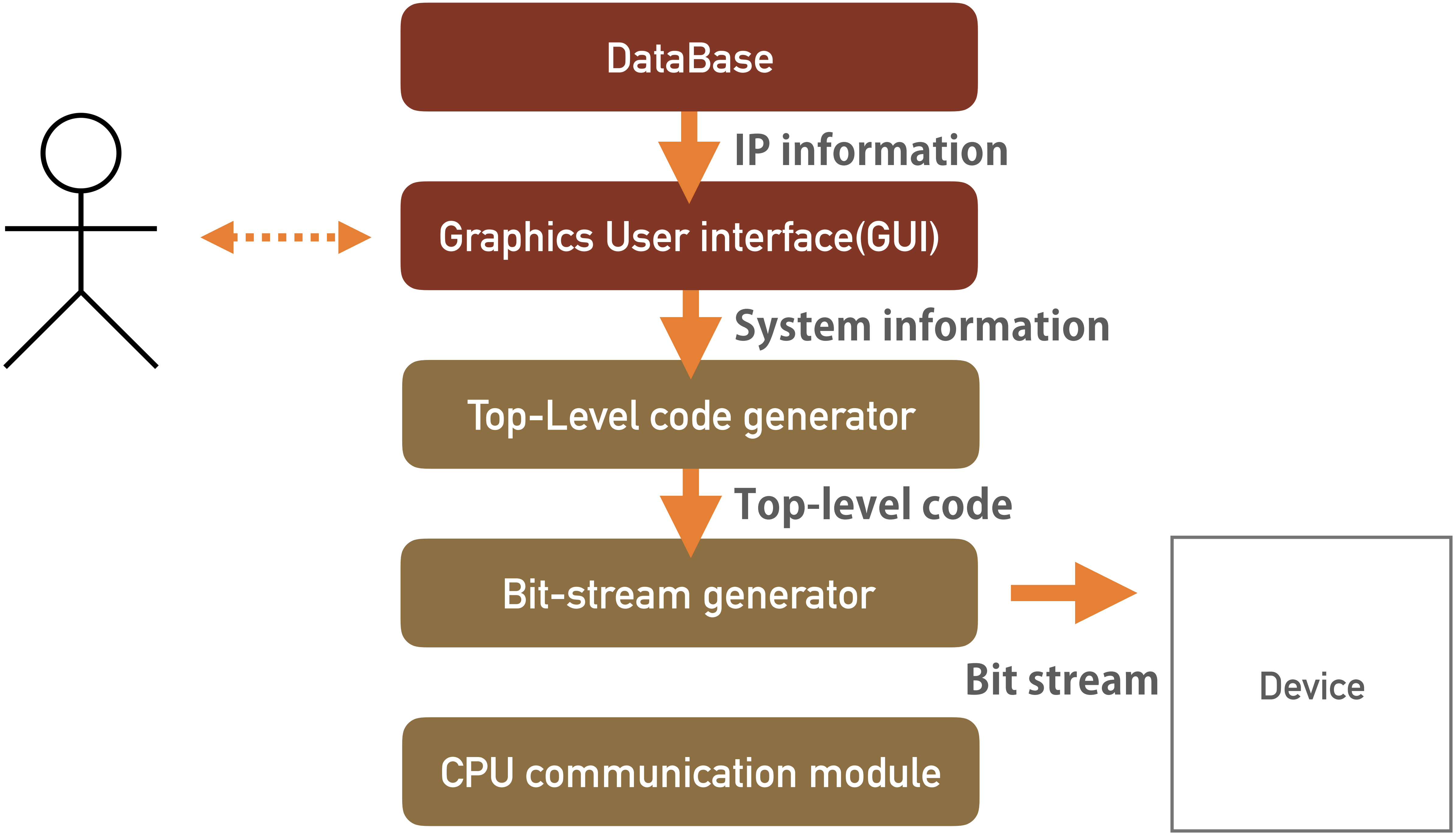
Top-Level code generator

Top-level code

Bit-stream generator

CPU communication module

INTRODUCTION – WORKFLOW



DataBase

IP information

Graphics User interface(GUI)

System information

Top-Level code generator

Top-level code

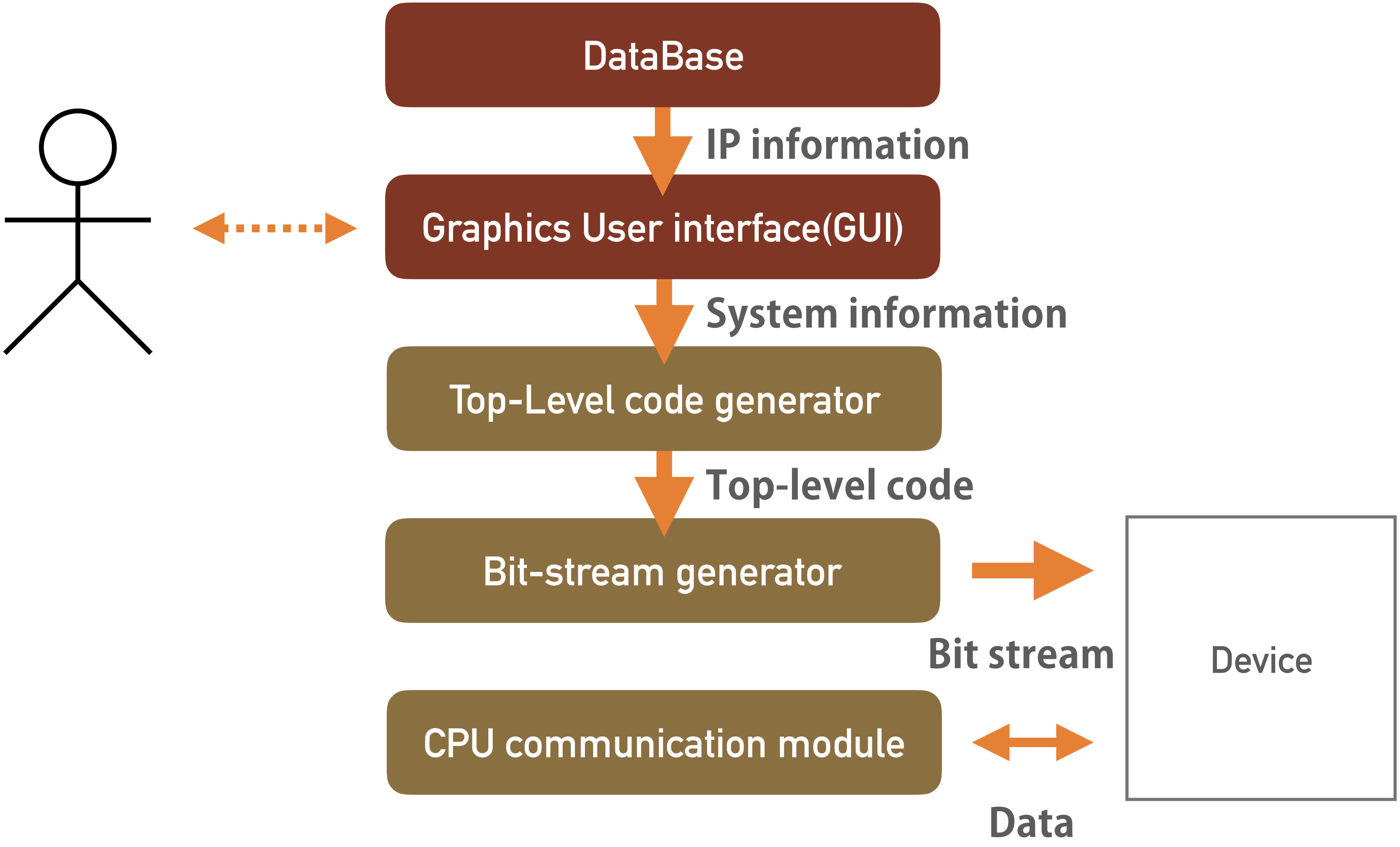
Bit-stream generator

Bit stream

Device

CPU communication module

INTRODUCTION – WORKFLOW



DataBase

IP information

Graphics User interface(GUI)

System information

Top-Level code generator

Top-level code

Bit-stream generator

Bit stream

CPU communication module

Data

Device

INTRODUCTION – WORK COOPERATION

DataBase

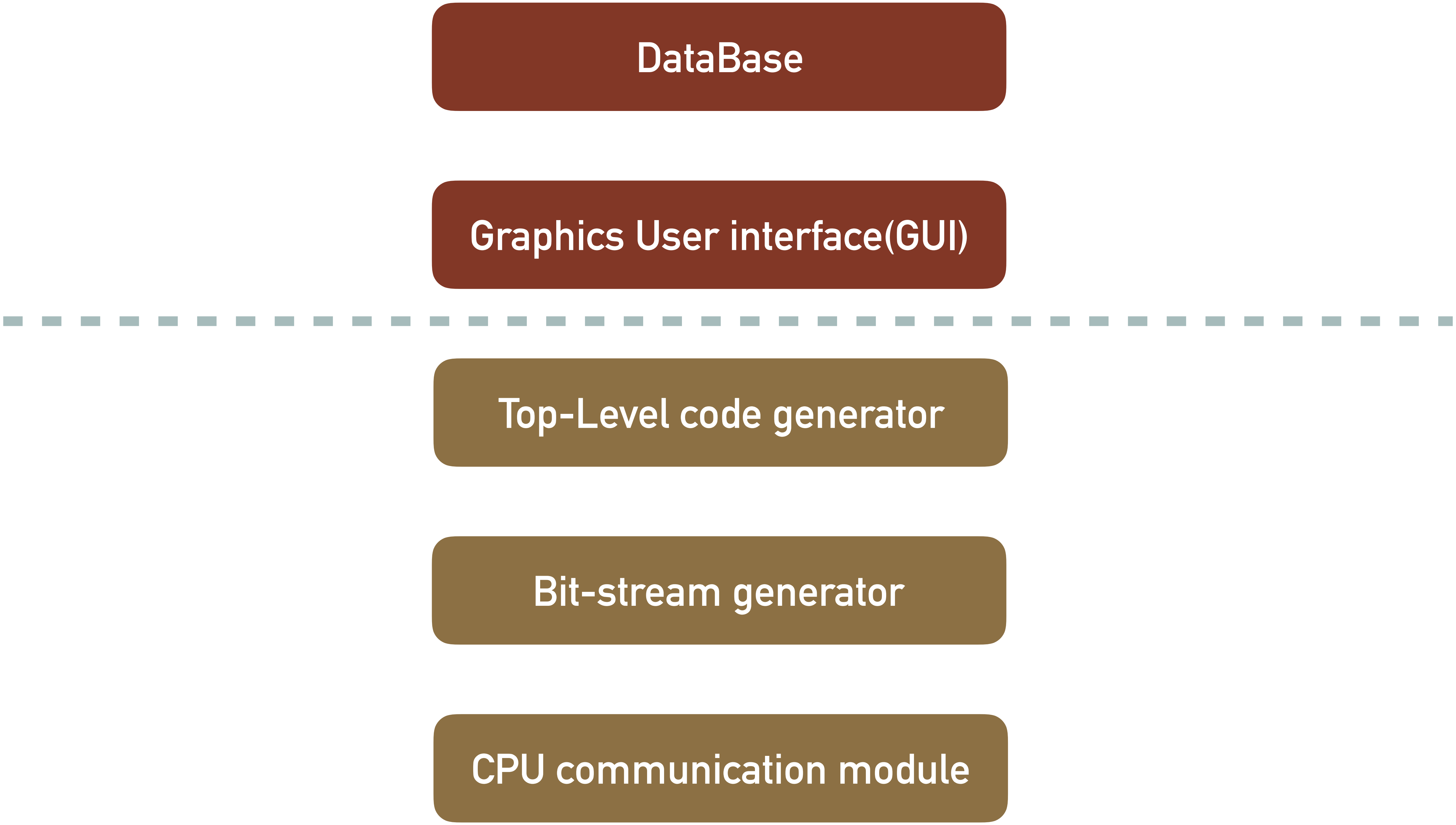
Graphics User interface(GUI)

Top-Level code generator

Bit-stream generator

CPU communication module

INTRODUCTION – WORK COOPERATION



DataBase

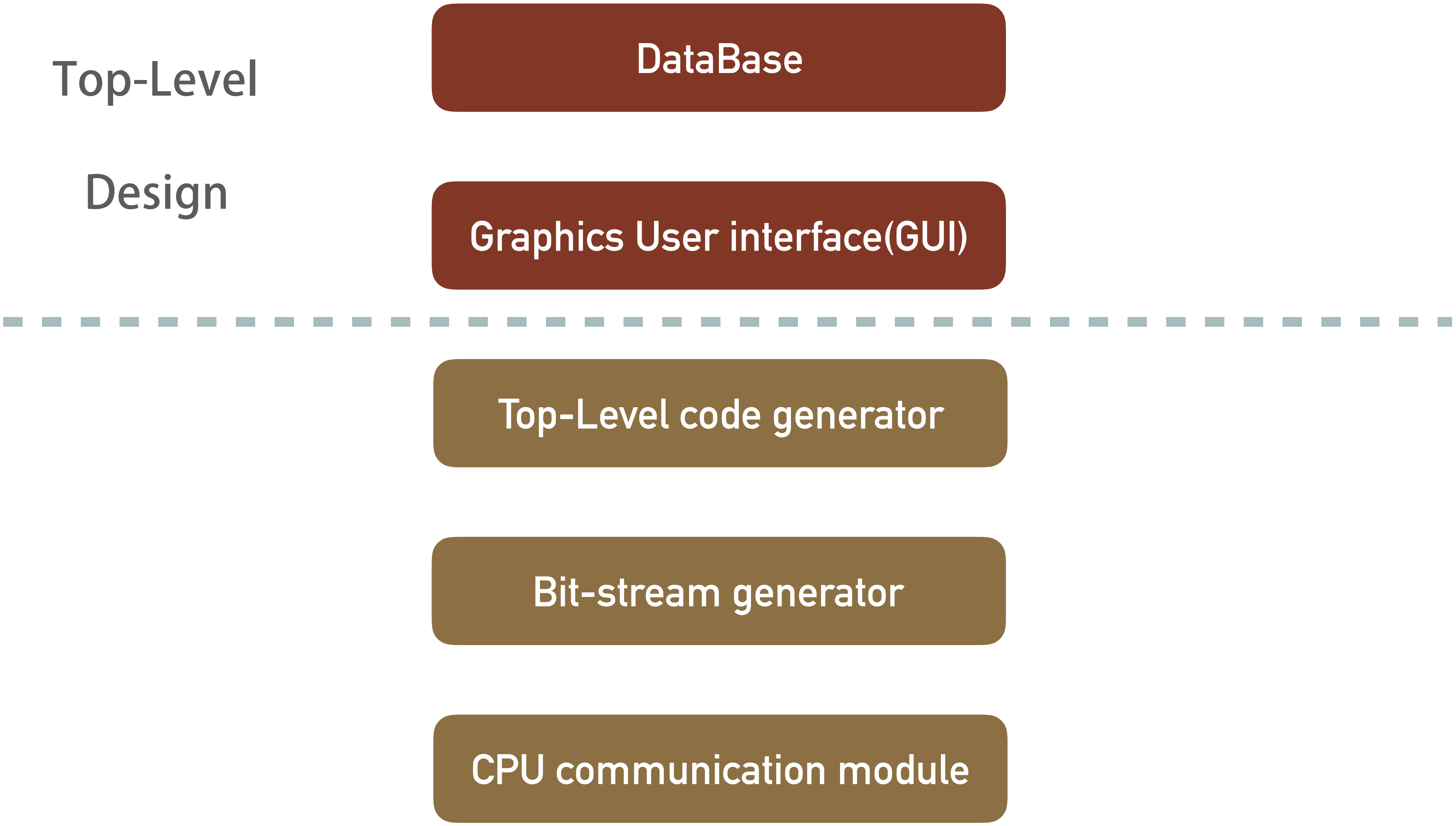
Graphics User interface(GUI)

Top-Level code generator

Bit-stream generator

CPU communication module

INTRODUCTION – WORK COOPERATION



Top-Level Design

DataBase

Graphics User interface(GUI)

Top-Level code generator

Bit-stream generator

CPU communication module

INTRODUCTION – WORK COOPERATION

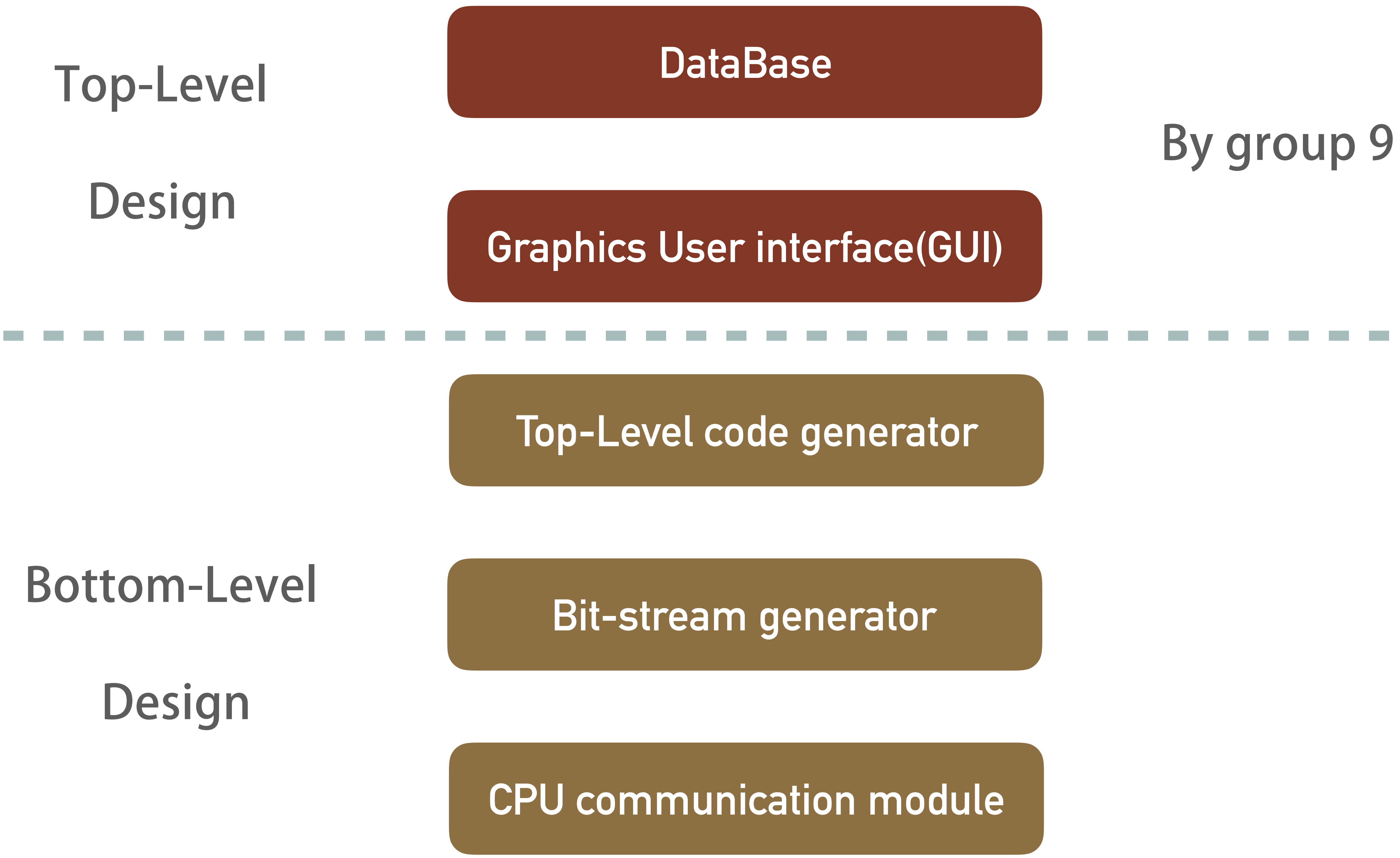
Top-Level Design

- DataBase
- Graphics User interface(GUI)

Bottom-Level Design

- Top-Level code generator
- Bit-stream generator
- CPU communication module

INTRODUCTION – WORK COOPERATION



Top-Level Design

DataBase

Graphics User interface(GUI)

By group 9

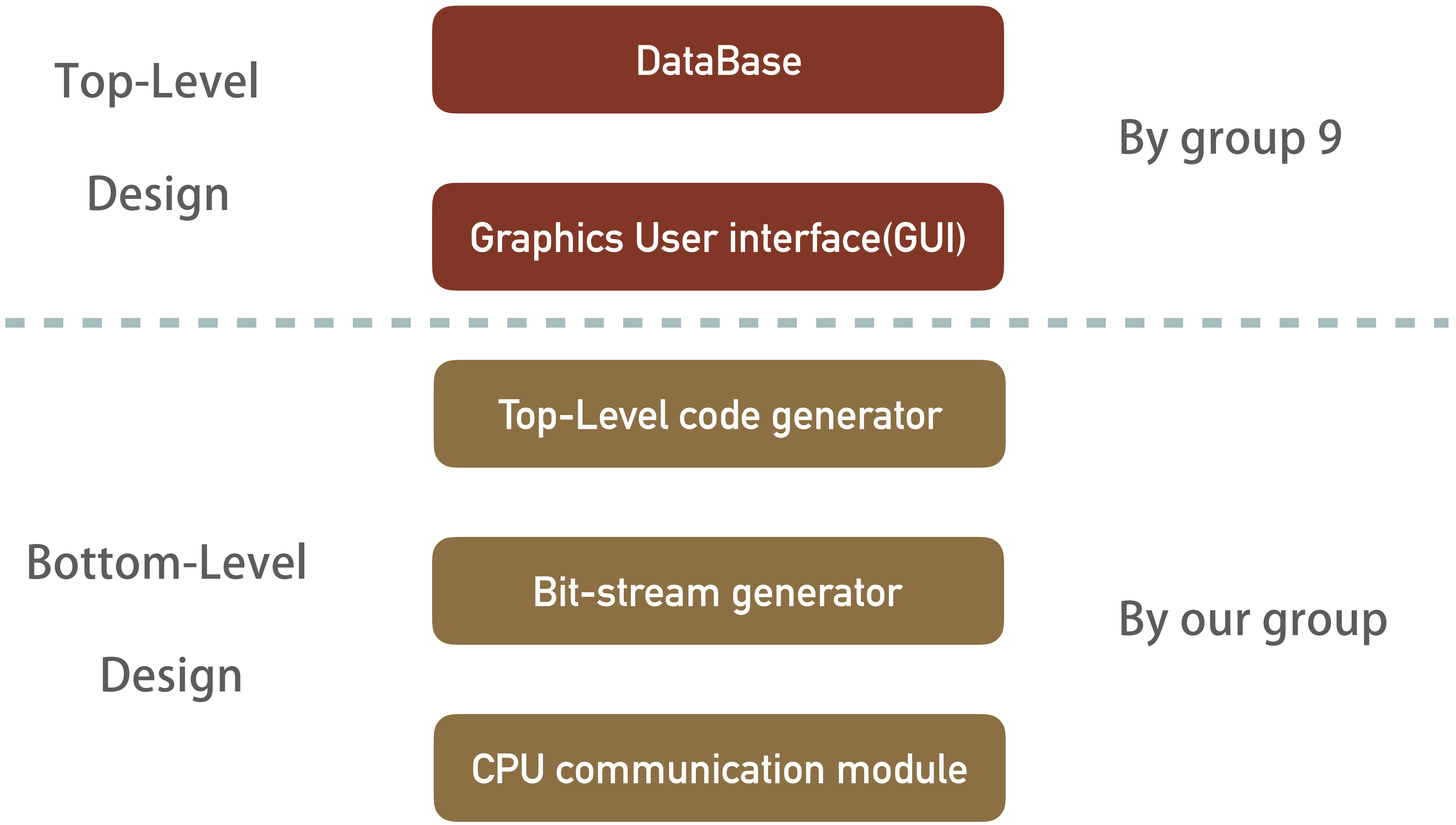
Bottom-Level Design

Top-Level code generator

Bit-stream generator

CPU communication module

INTRODUCTION – WORK COOPERATION



Top-Level Design

DataBase

Graphics User interface(GUI)

By group 9

Bottom-Level Design

Top-Level code generator

Bit-stream generator

CPU communication module

By our group

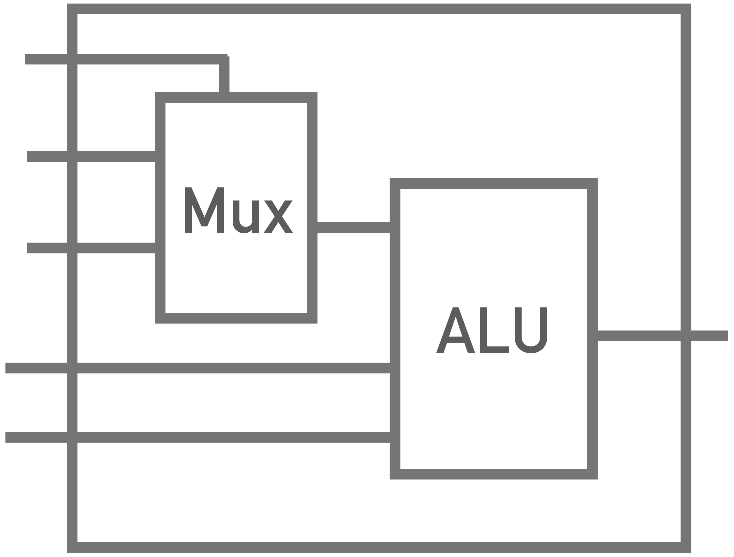
OUTLINE

- Introduction
- **Top Level code generator**
- Bitstream generator
- CPU communication module
- Summary

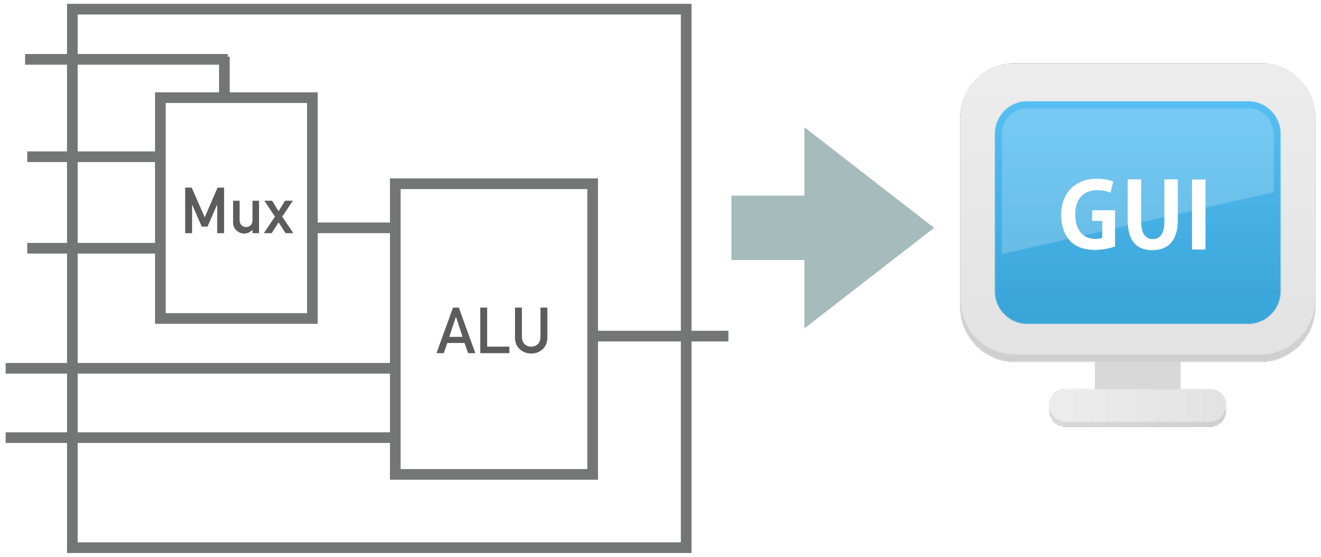
TOP LEVEL CODE GENERATOR

- Top-level entity generator is designed for generating top-level entity VHDL codes of the entire system by the system information collected by GUI.
- This module is supposed to be used when user finished constructing the entire system.

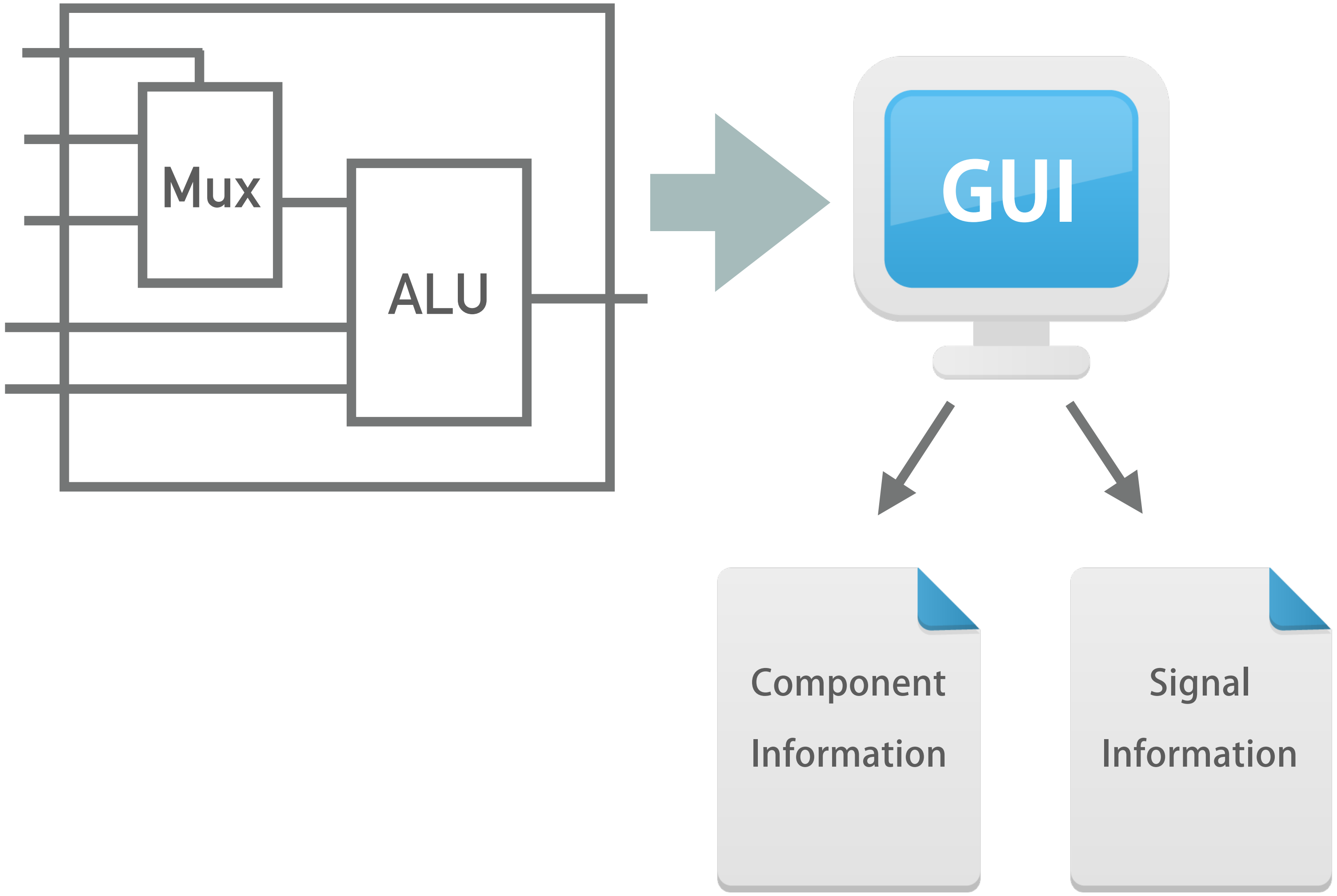
TOP LEVEL CODE GENERATOR - HOW DOES IT WORK



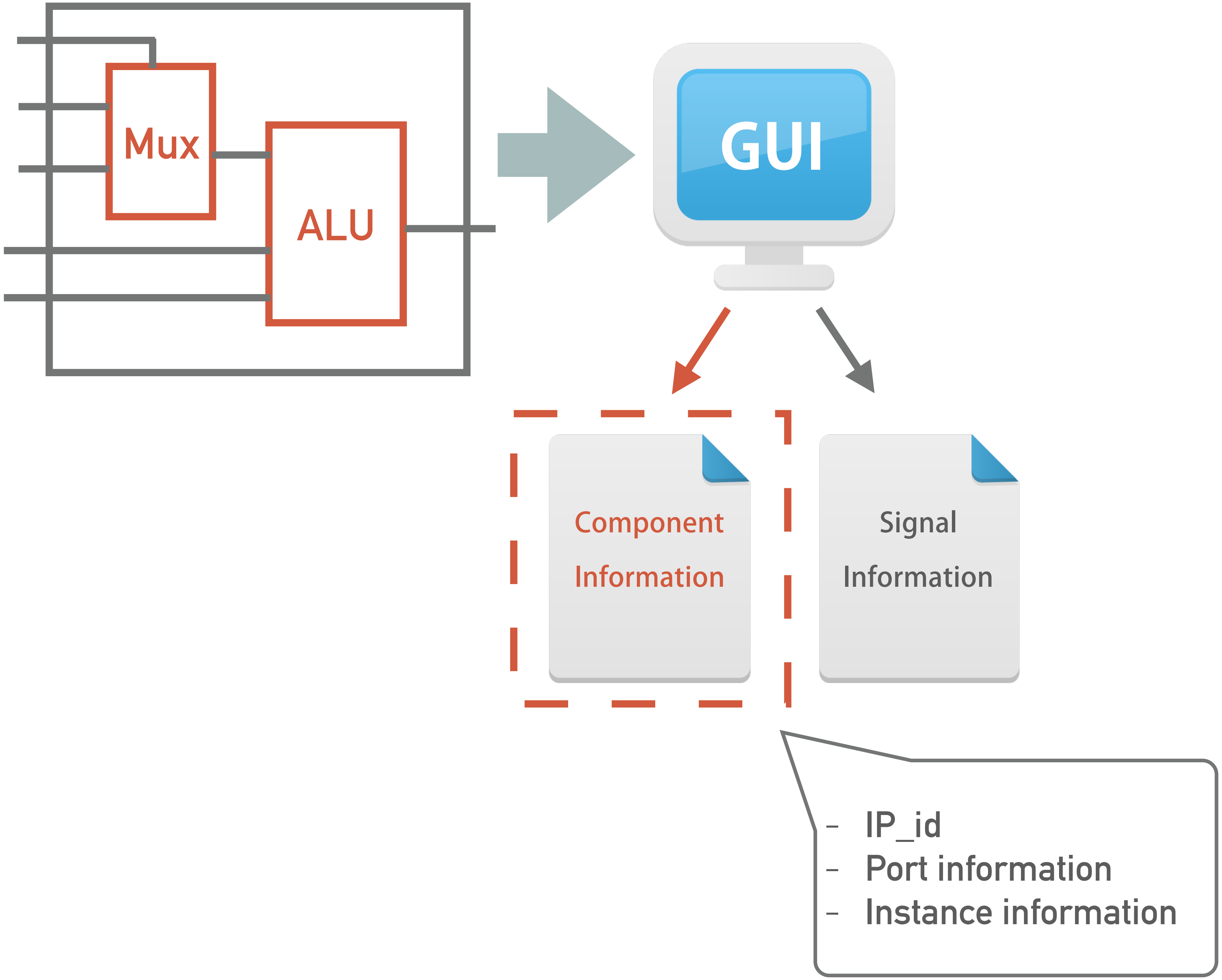
TOP LEVEL CODE GENERATOR – HOW DOES IT WORK



TOP LEVEL CODE GENERATOR – HOW DOES IT WORK



TOP LEVEL CODE GENERATOR - HOW DOES IT WORK

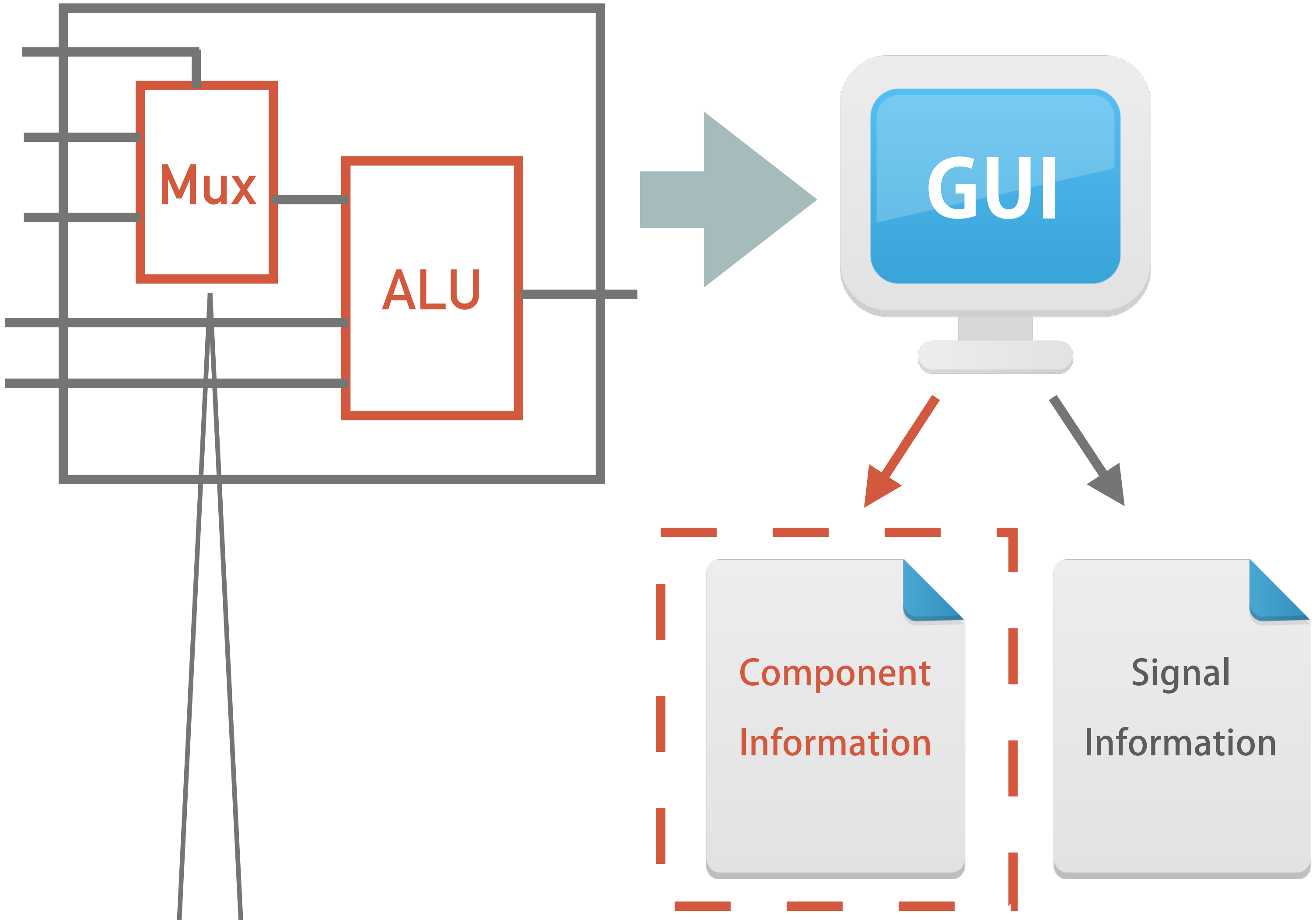


Component Information

Signal Information

- IP_id
- Port information
- Instance information

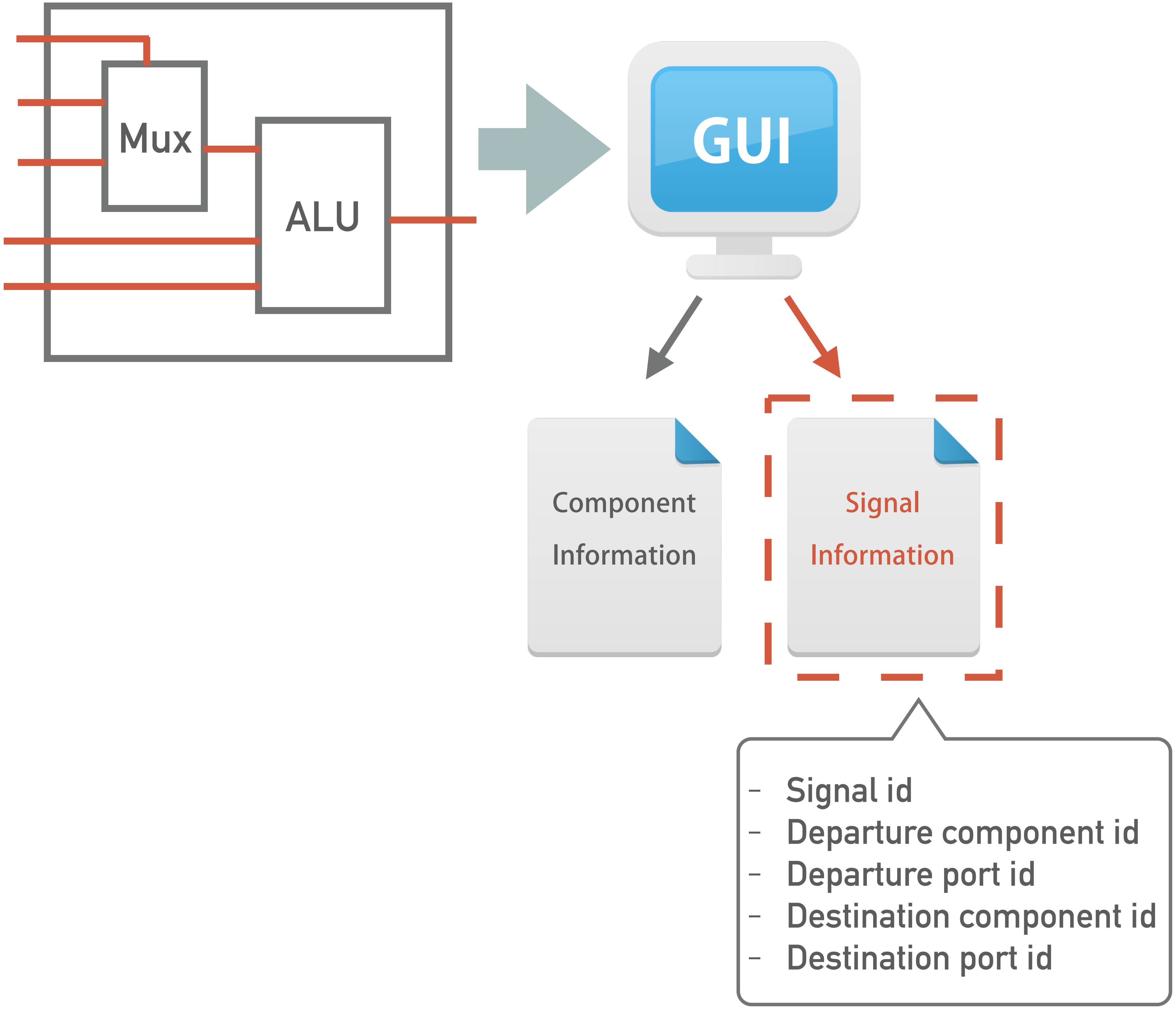
TOP LEVEL CODE GENERATOR – HOW DOES IT WORK



```
1 Mux 3 1
data_1    std_logic_vector(7 downto 0) 0 in
data_2    std_logic_vector(7 downto 0) 1 in
sel        std_logic                   2 in
data_out std_logic_vector(7 downto 0) 3 out
1
```

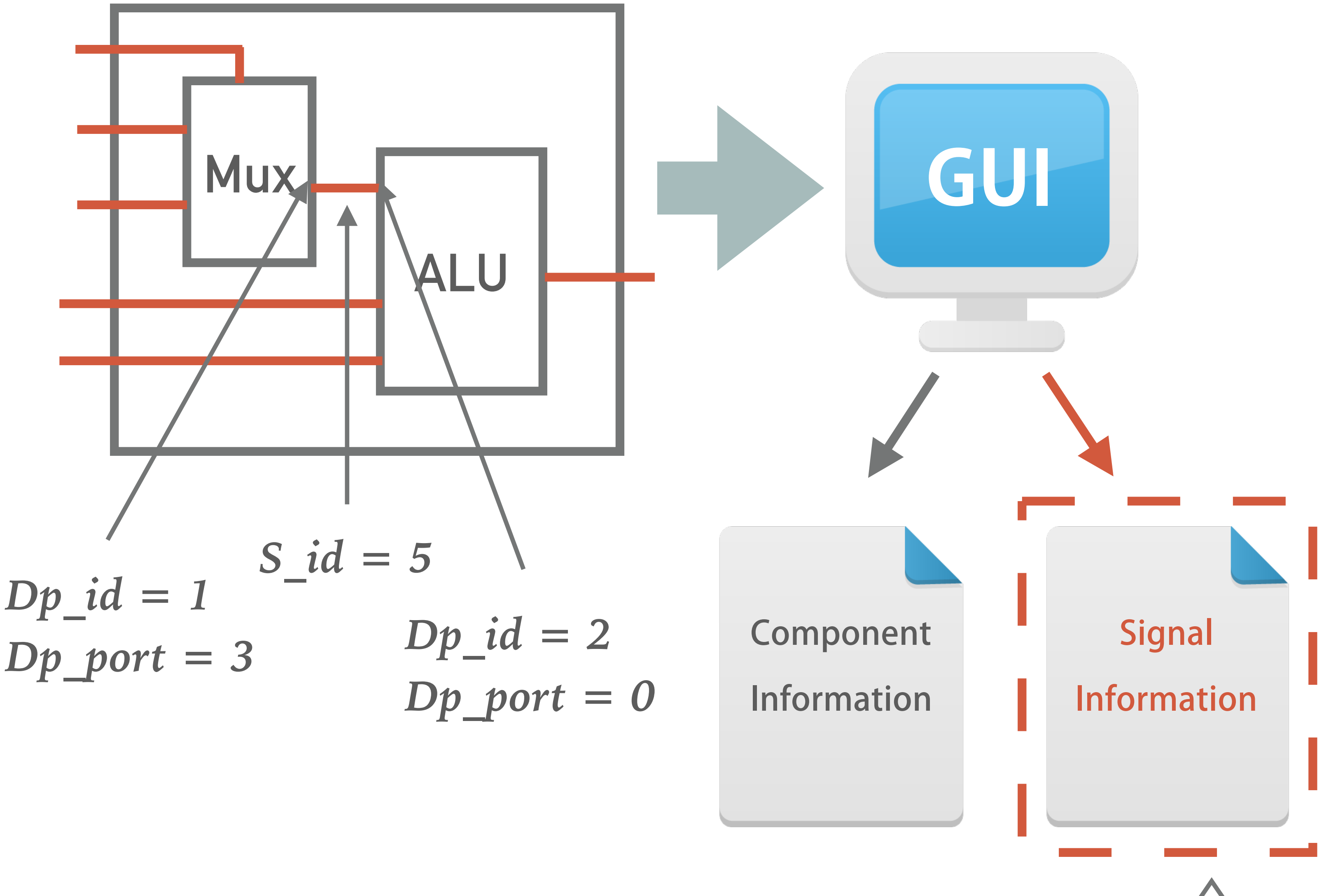
- IP_id
- Port information
- Instance information

TOP LEVEL CODE GENERATOR – HOW DOES IT WORK



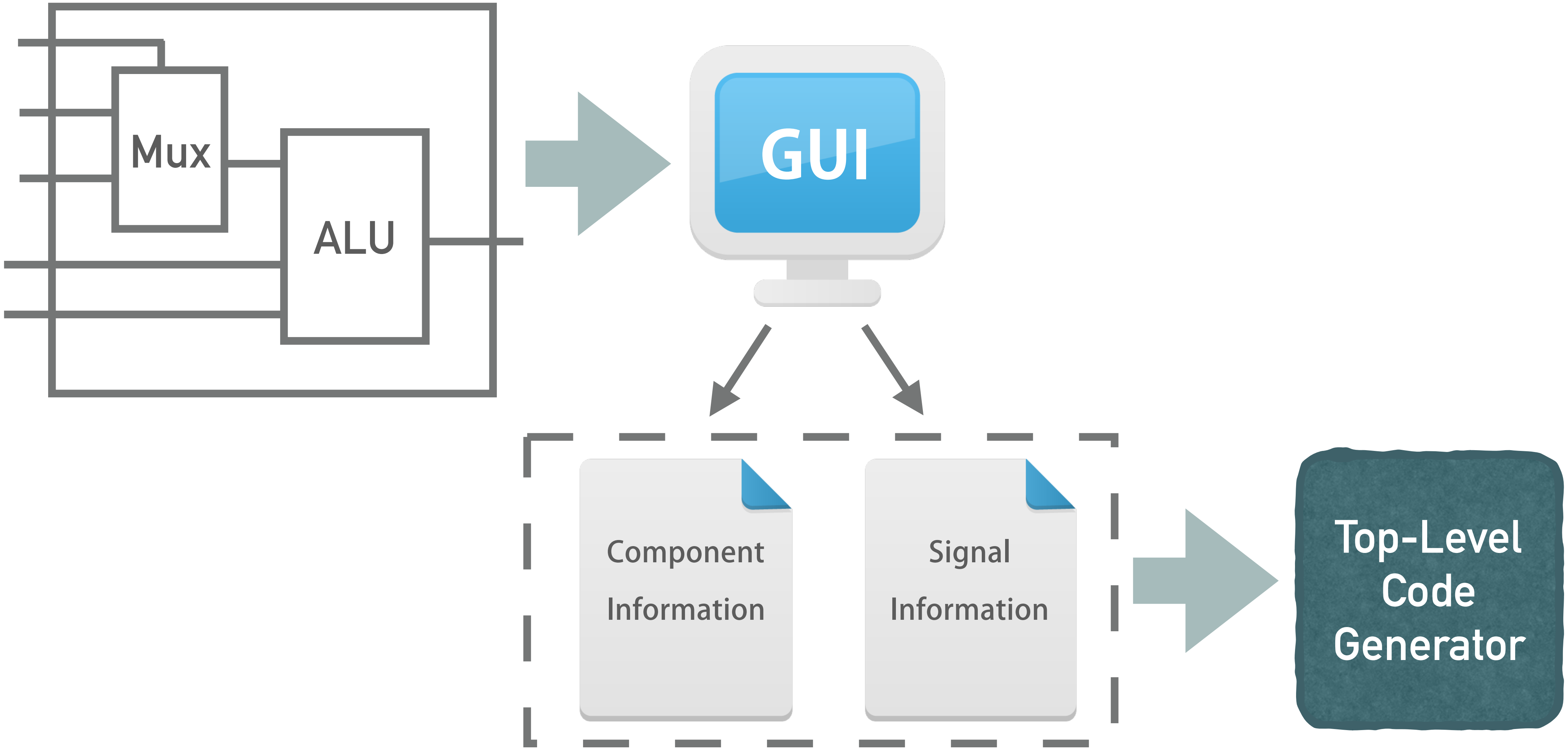
- Signal id
- Departure component id
- Departure port id
- Destination component id
- Destination port id

TOP LEVEL CODE GENERATOR – HOW DOES IT WORK

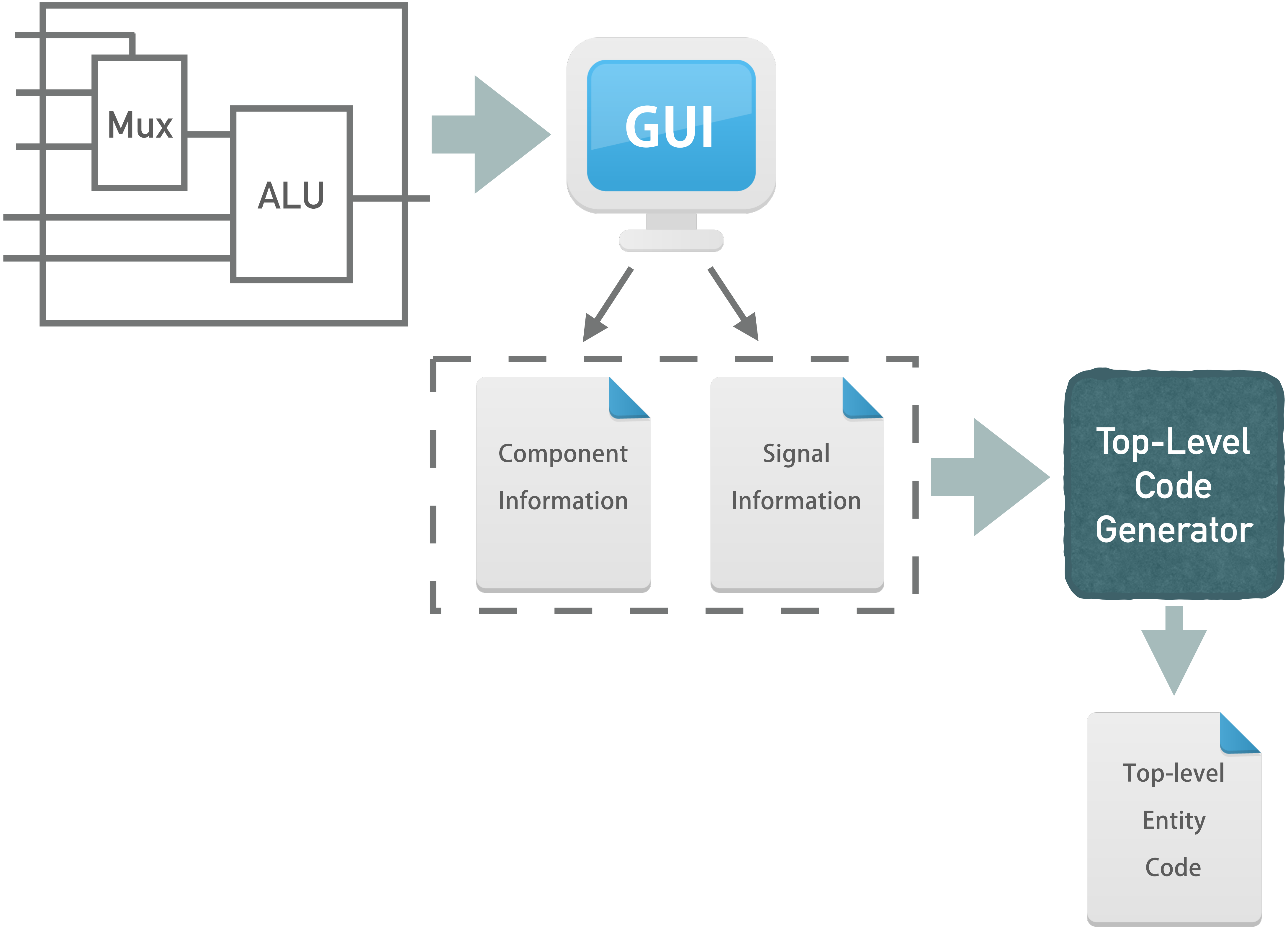


- Signal id
- Departure component id
- Departure port id
- Destination component id
- Destination port id

TOP LEVEL CODE GENERATOR - HOW DOES IT WORK



TOP LEVEL CODE GENERATOR – HOW DOES IT WORK



Mux

ALU

GUI

Component Information

Signal Information

Top-Level Code Generator

Top-level Entity Code

OUTLINE

- Introduction
- Top-Level code generator
- **Bitstream generator**
- CPU communication module
- Summary

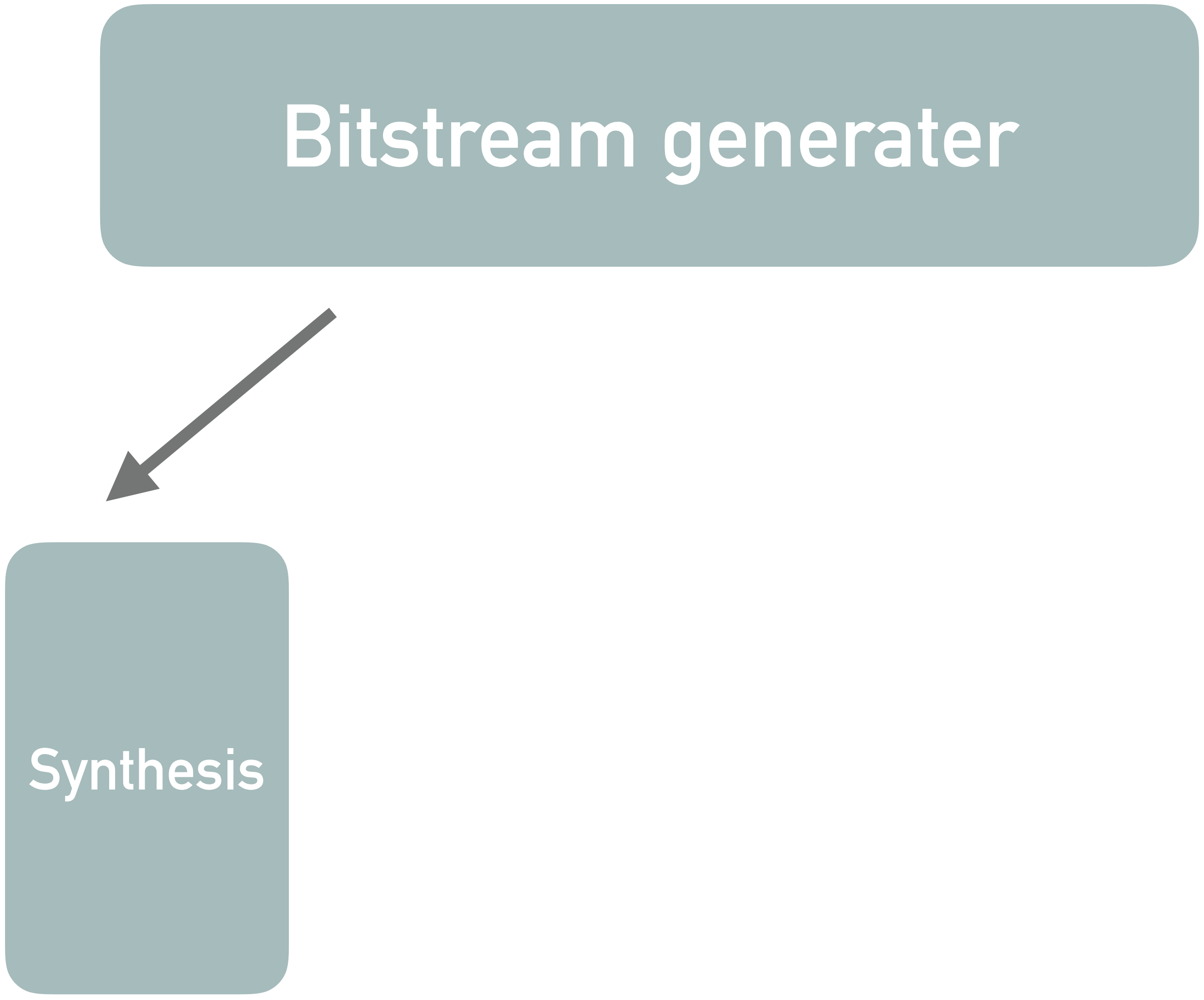
BITSTREAM GENERATOR

- Based on the function of Diamond Lattice
- To accomplish this function we learn to how to generate the bitstream in Diamond Lattice manually.
- Then we develop a program to automate this progress

BITSTREAM GENERATOR – HOW DOES IT WORK

Bitstream generater

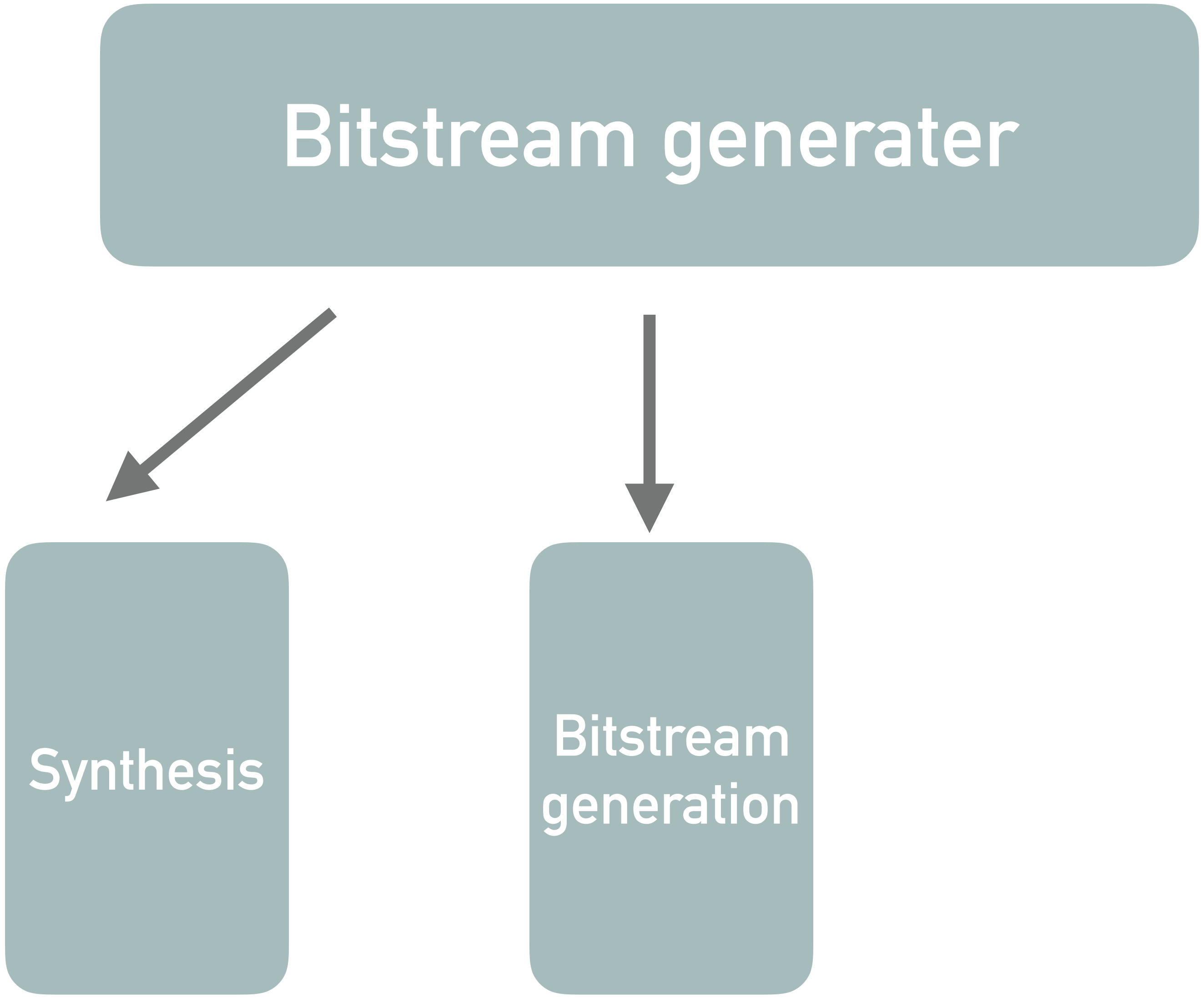
BITSTREAM GENERATOR - HOW DOES IT WORK



Bitstream generater

Synthesis

BITSTREAM GENERATOR - HOW DOES IT WORK

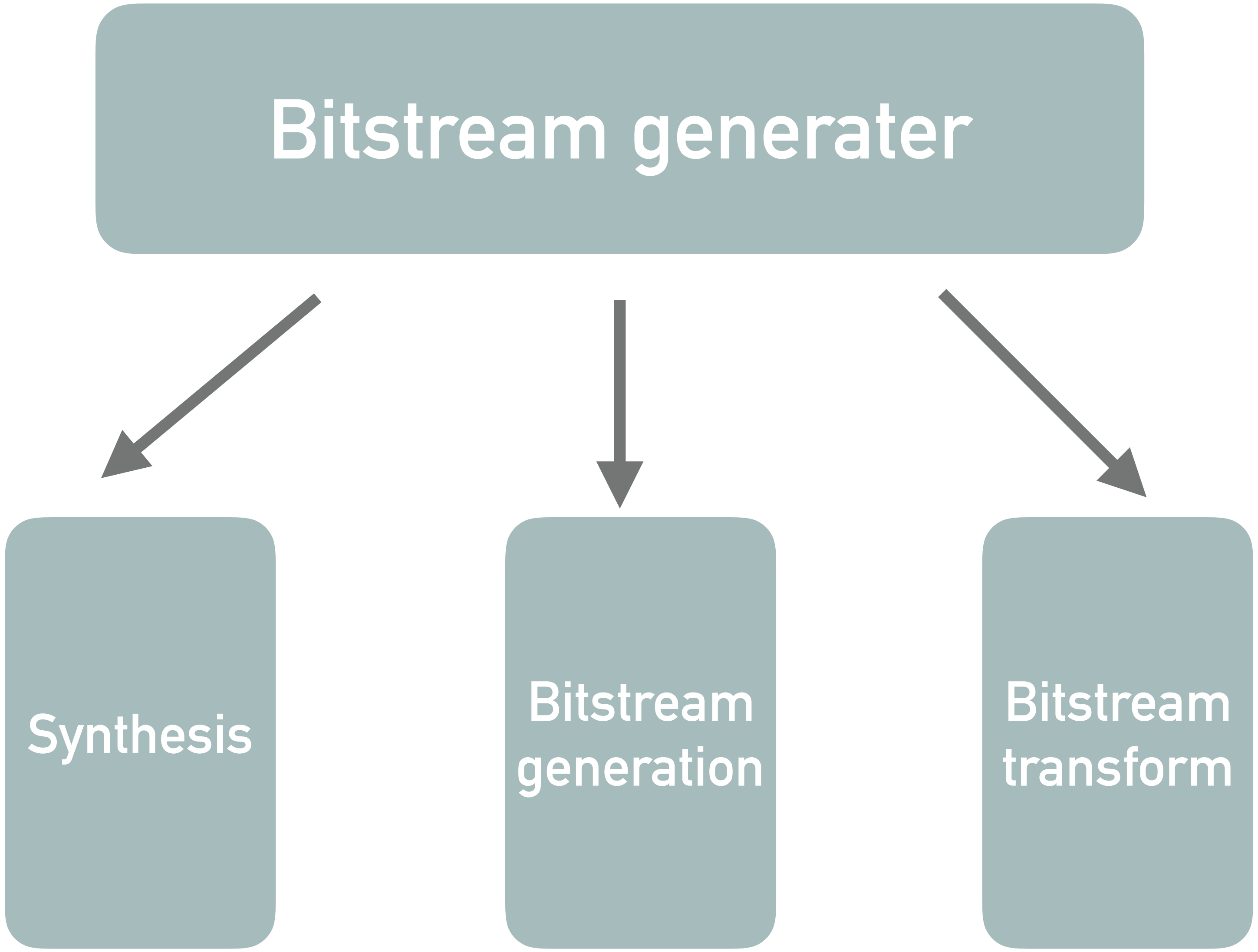


Bitstream generater

Synthesis

Bitstream generation

BITSTREAM GENERATOR - HOW DOES IT WORK



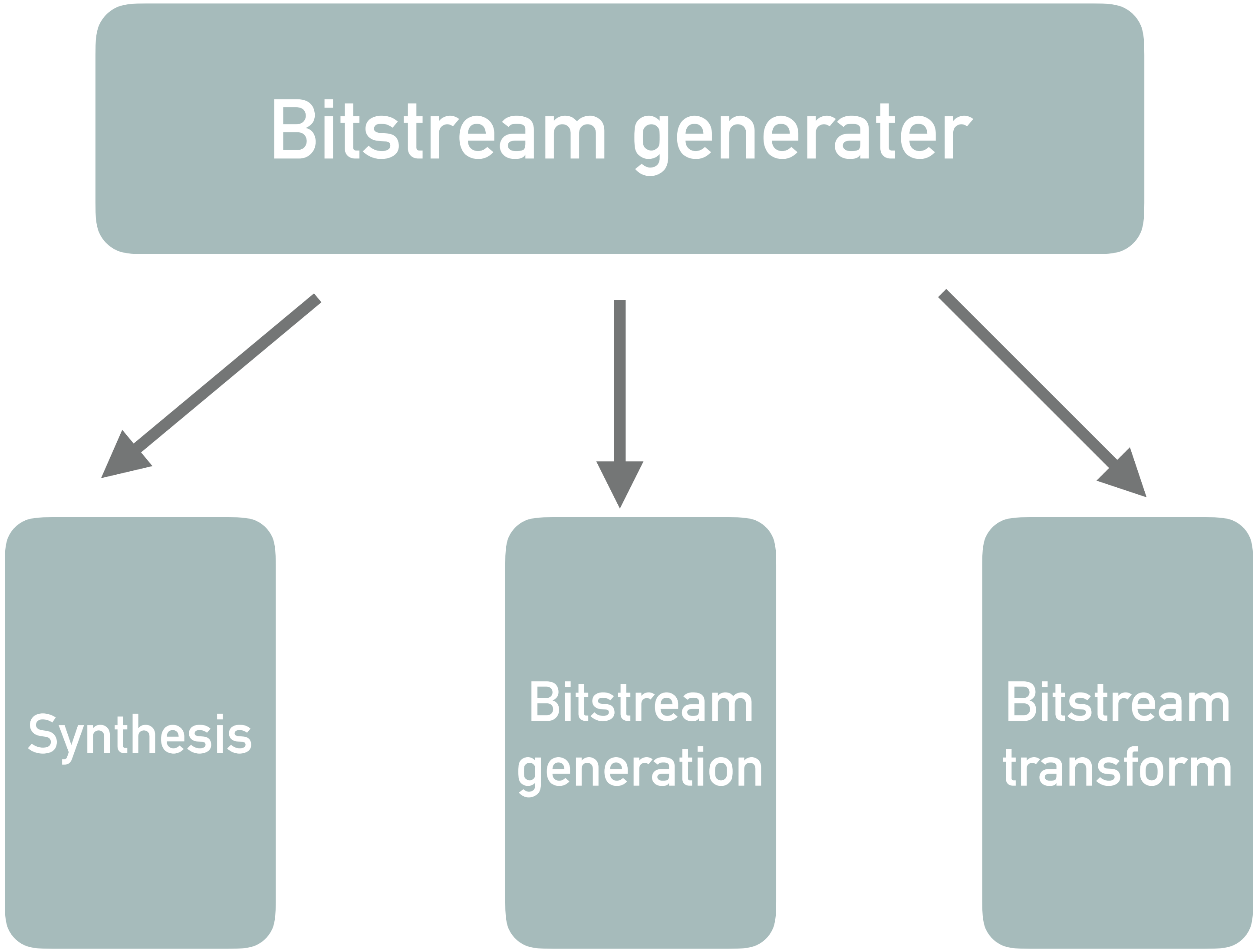
Bitstream generater

Synthesis

Bitstream generation

Bitstream transform

BITSTREAM GENERATOR - HOW DOES IT WORK



Bitstream generater

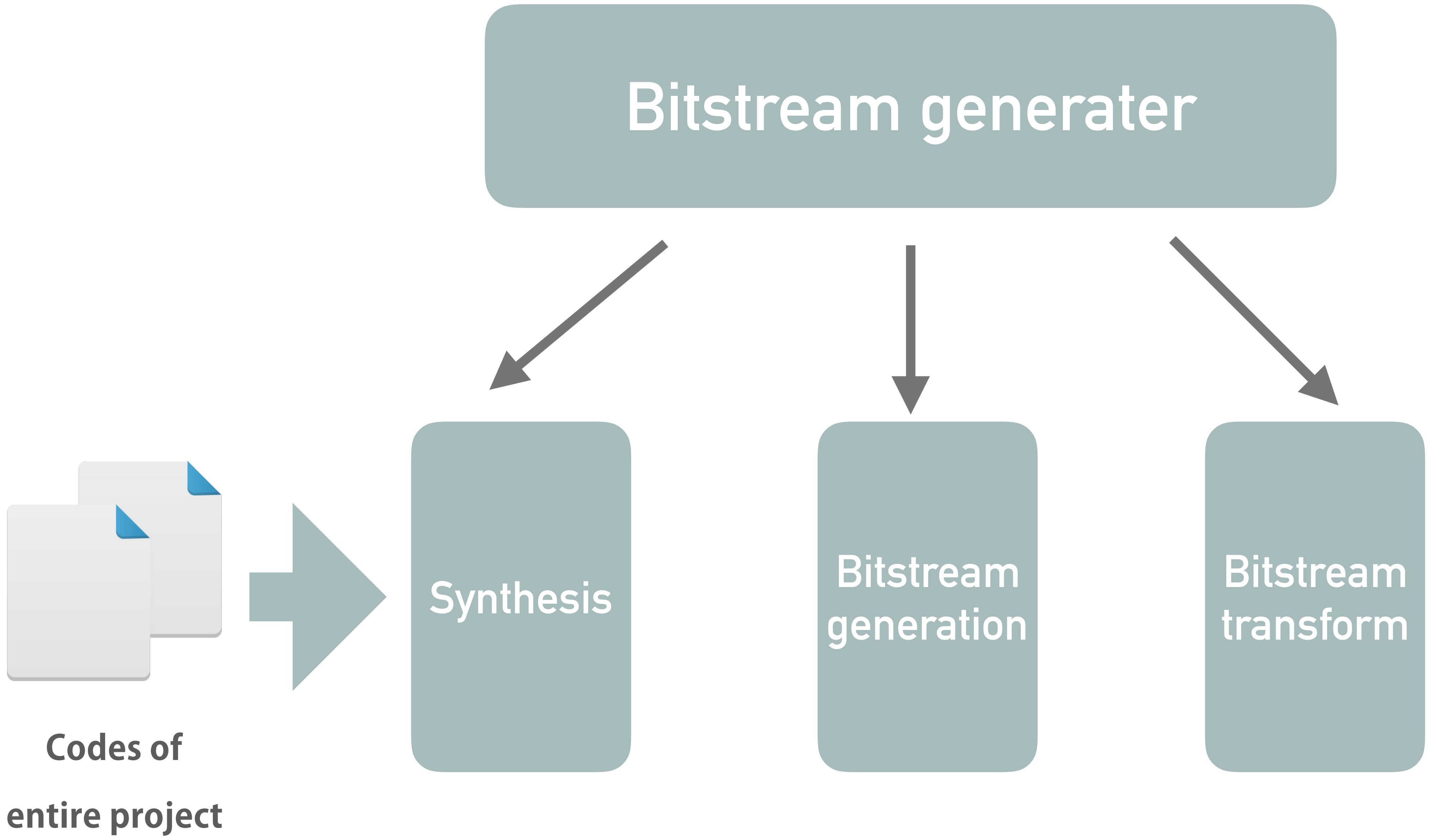
Synthesis

Bitstream generation

Bitstream transform

Codes of entire project

BITSTREAM GENERATOR – HOW DOES IT WORK



Bitstream generater

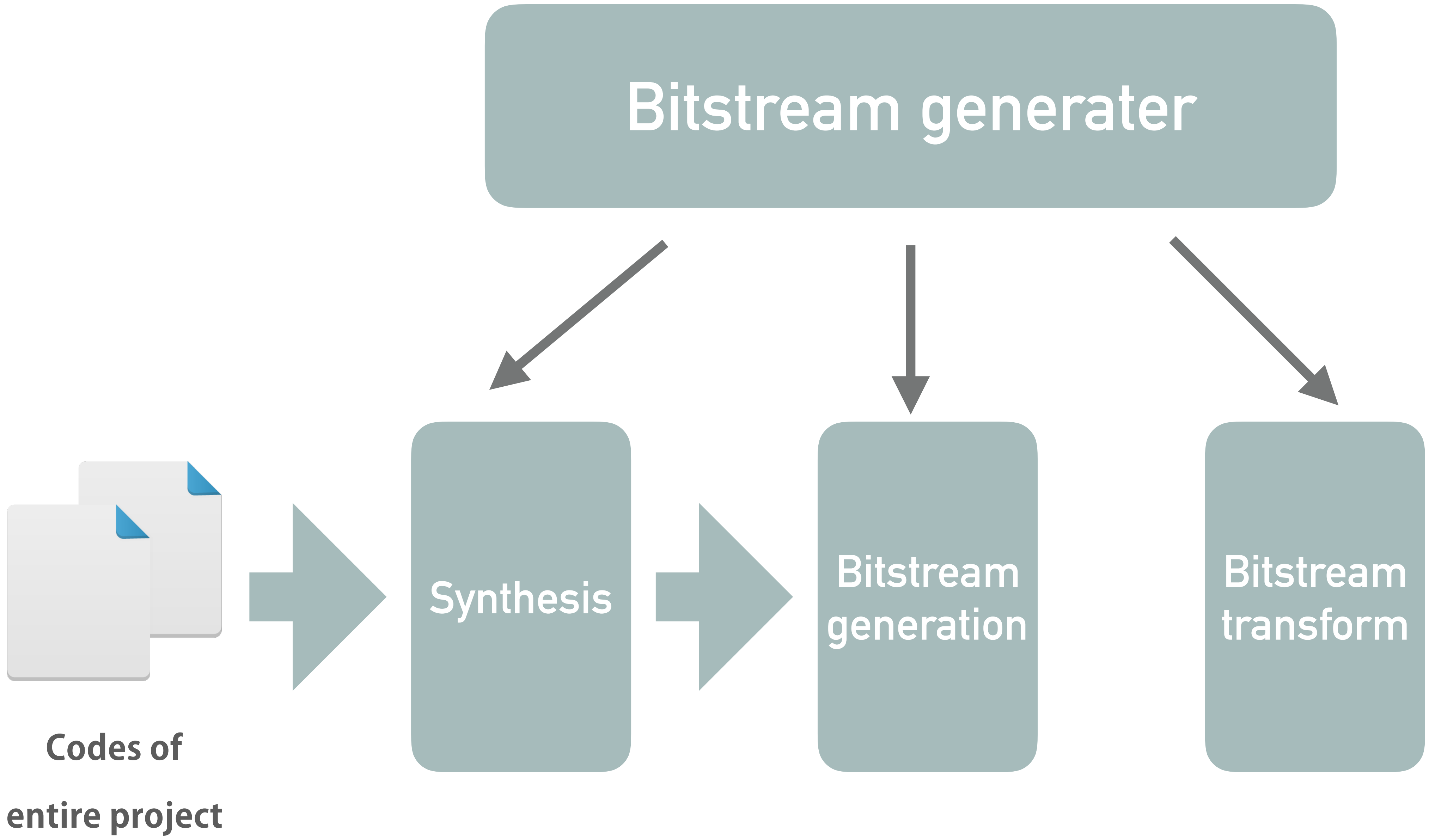
Synthesis

Bitstream generation

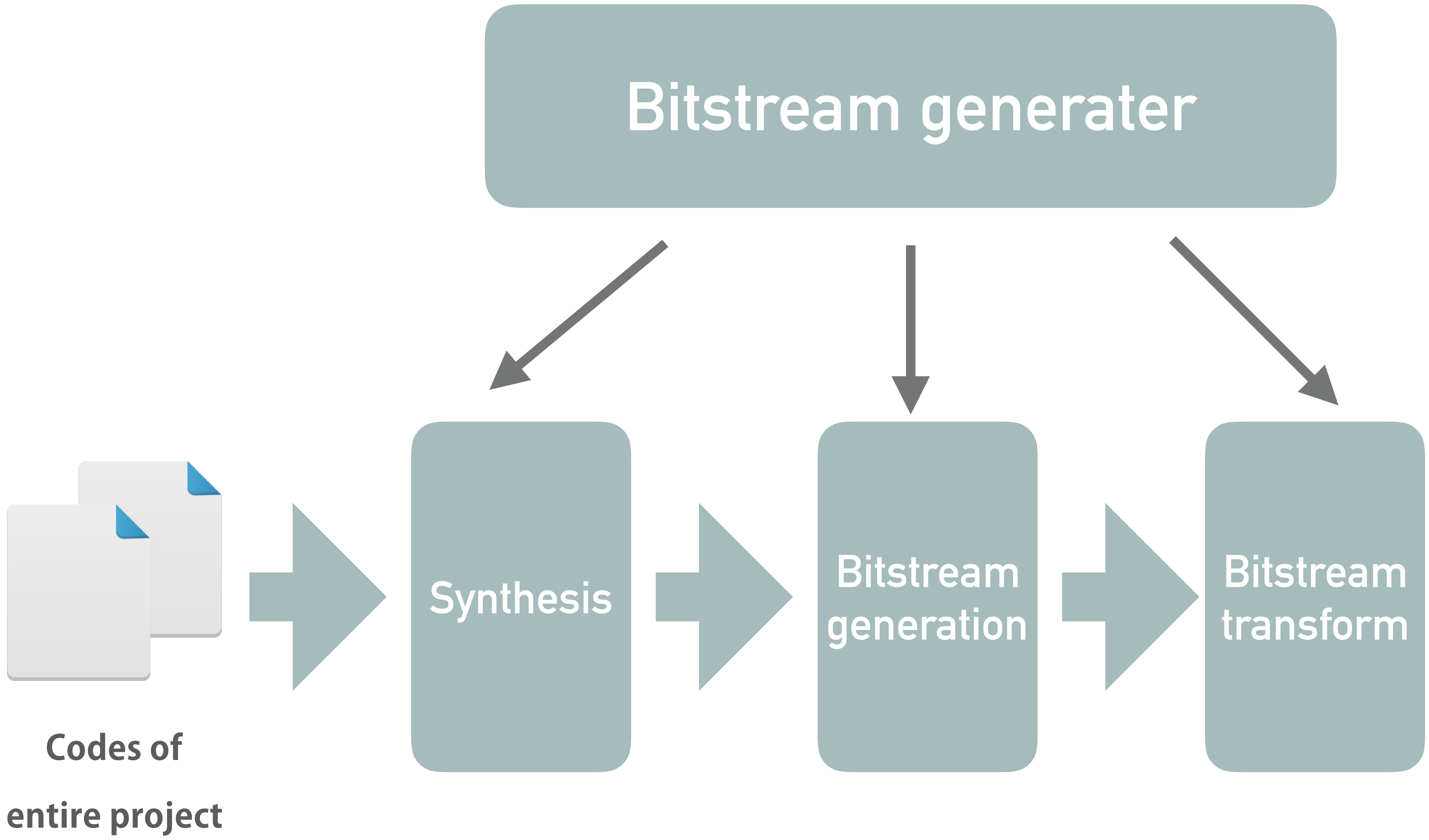
Bitstream transform

Codes of entire project

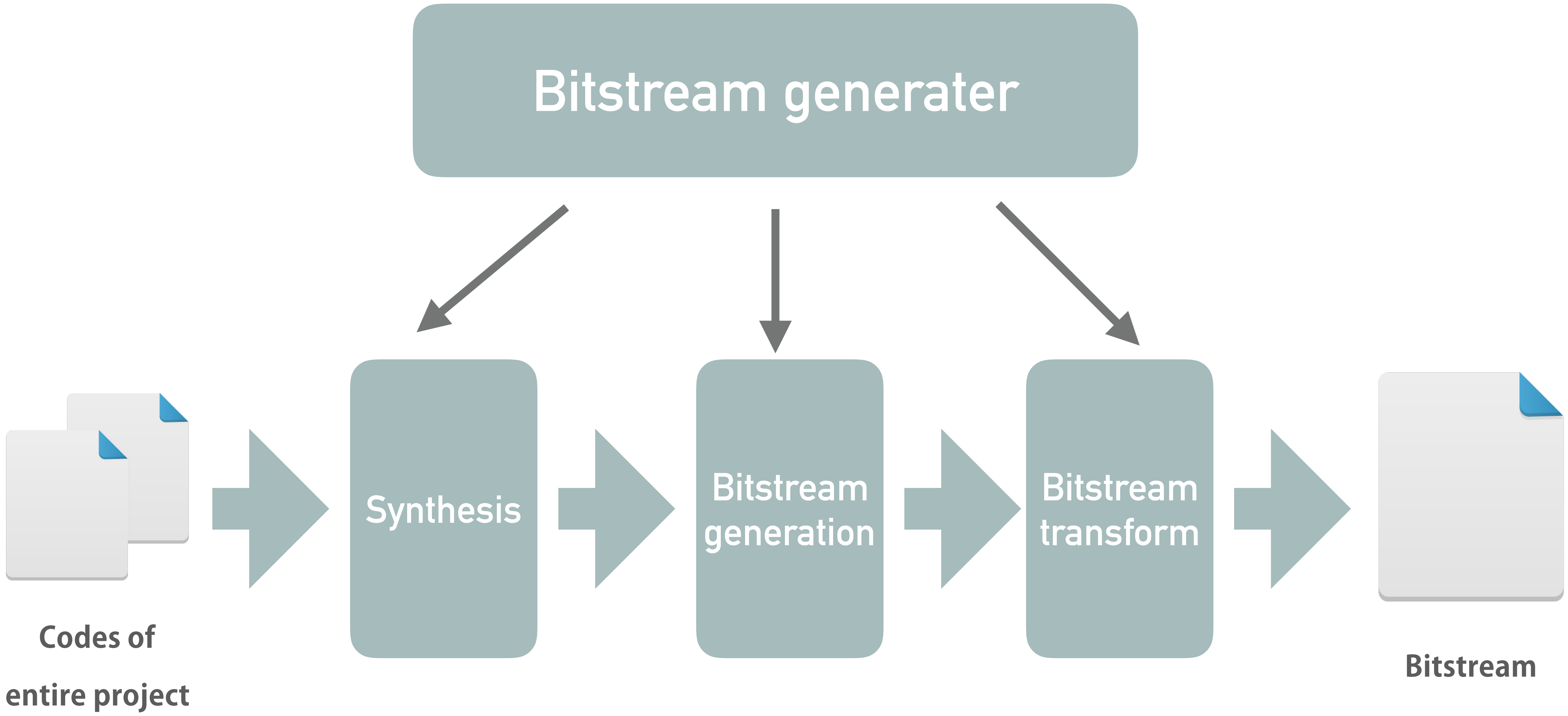
BITSTREAM GENERATOR – HOW DOES IT WORK



BITSTREAM GENERATOR – HOW DOES IT WORK



BITSTREAM GENERATOR – HOW DOES IT WORK



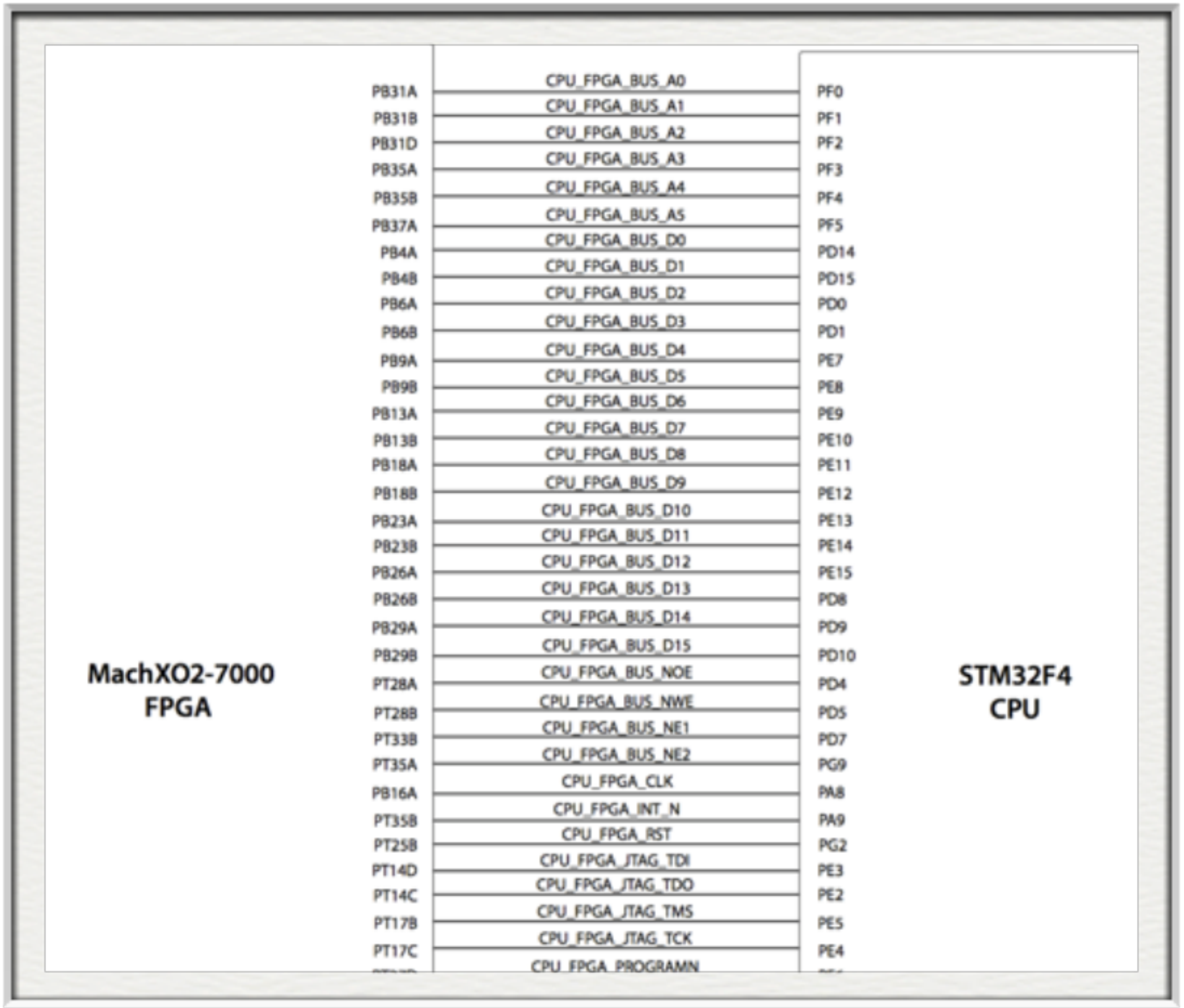
OUTLINE

- Introduction
- Top-Level code generator
- Bitstream generator
- **CPU communication module**
- Summary

CPU COMMUNICATION MODULE

- Transmit data between CPU and FPGA in run-time
- It provides a small library which contains functions to manage data transmission between CPU and FPGA
- Since in our project, we don’t need to do anything about simulation, GUI won’t provide the interface to invoke this module.
- It could be a useful module for the future work.

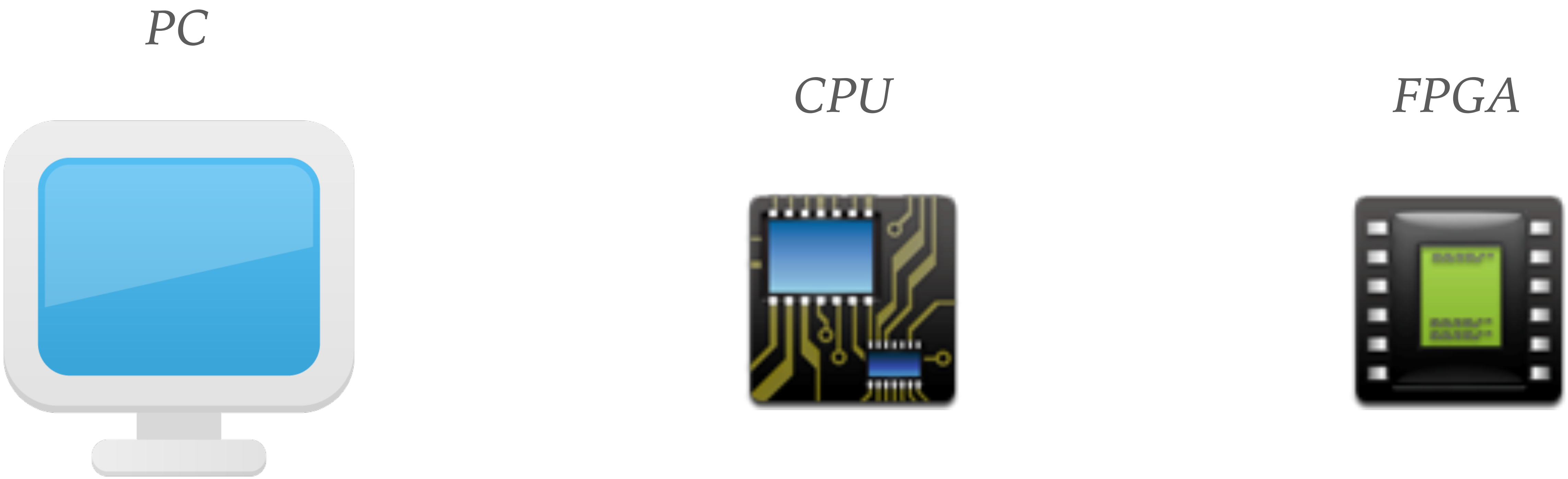
CPU COMMUNICATION MODULE – WHY WE NEED THIS MODULE



PB31A	CPU_FPGA_BUS_A0	PF0
PB31B	CPU_FPGA_BUS_A1	PF1
PB31D	CPU_FPGA_BUS_A2	PF2
PB35A	CPU_FPGA_BUS_A3	PF3
PB35B	CPU_FPGA_BUS_A4	PF4
PB37A	CPU_FPGA_BUS_A5	PF5
PB4A	CPU_FPGA_BUS_D0	PD14
PB4B	CPU_FPGA_BUS_D1	PD15
PB6A	CPU_FPGA_BUS_D2	PD0
PB6B	CPU_FPGA_BUS_D3	PD1
PB9A	CPU_FPGA_BUS_D4	PE7
PB9B	CPU_FPGA_BUS_D5	PE8
PB13A	CPU_FPGA_BUS_D6	PE9
PB13B	CPU_FPGA_BUS_D7	PE10
PB18A	CPU_FPGA_BUS_D8	PE11
PB18B	CPU_FPGA_BUS_D9	PE12
PB23A	CPU_FPGA_BUS_D10	PE13
PB23B	CPU_FPGA_BUS_D11	PE14
PB26A	CPU_FPGA_BUS_D12	PE15
PB26B	CPU_FPGA_BUS_D13	PD8
PB29A	CPU_FPGA_BUS_D14	PD9
PB29B	CPU_FPGA_BUS_D15	PD10
PT28A	CPU_FPGA_BUS_NOE	PD4
PT28B	CPU_FPGA_BUS_NWE	PD5
PT33B	CPU_FPGA_BUS_NE1	PD7
PT35A	CPU_FPGA_BUS_NE2	PG9
PB16A	CPU_FPGA_CLK	PA8
PT35B	CPU_FPGA_INT_N	PA9
PT25B	CPU_FPGA_RST	PG2
PT14D	CPU_FPGA_JTAG_TDI	PE3
PT14C	CPU_FPGA_JTAG_TDO	PE2
PT17B	CPU_FPGA_JTAG_TMS	PE5
PT17C	CPU_FPGA_JTAG_TCK	PE4
	CPU_FPGA_PROGRAMN	

- ➤ We check the data sheet of SECUBE. We find CPU communicates with FPGA by GPIO.
- ➤ Till now there is not any function to handle data transmission between GPIO of CPU and input buffer of FPGA. User need to control each corresponding bit of GPIO. It is not convenient.

CPU COMMUNICATION MODULE – WORK FLOW

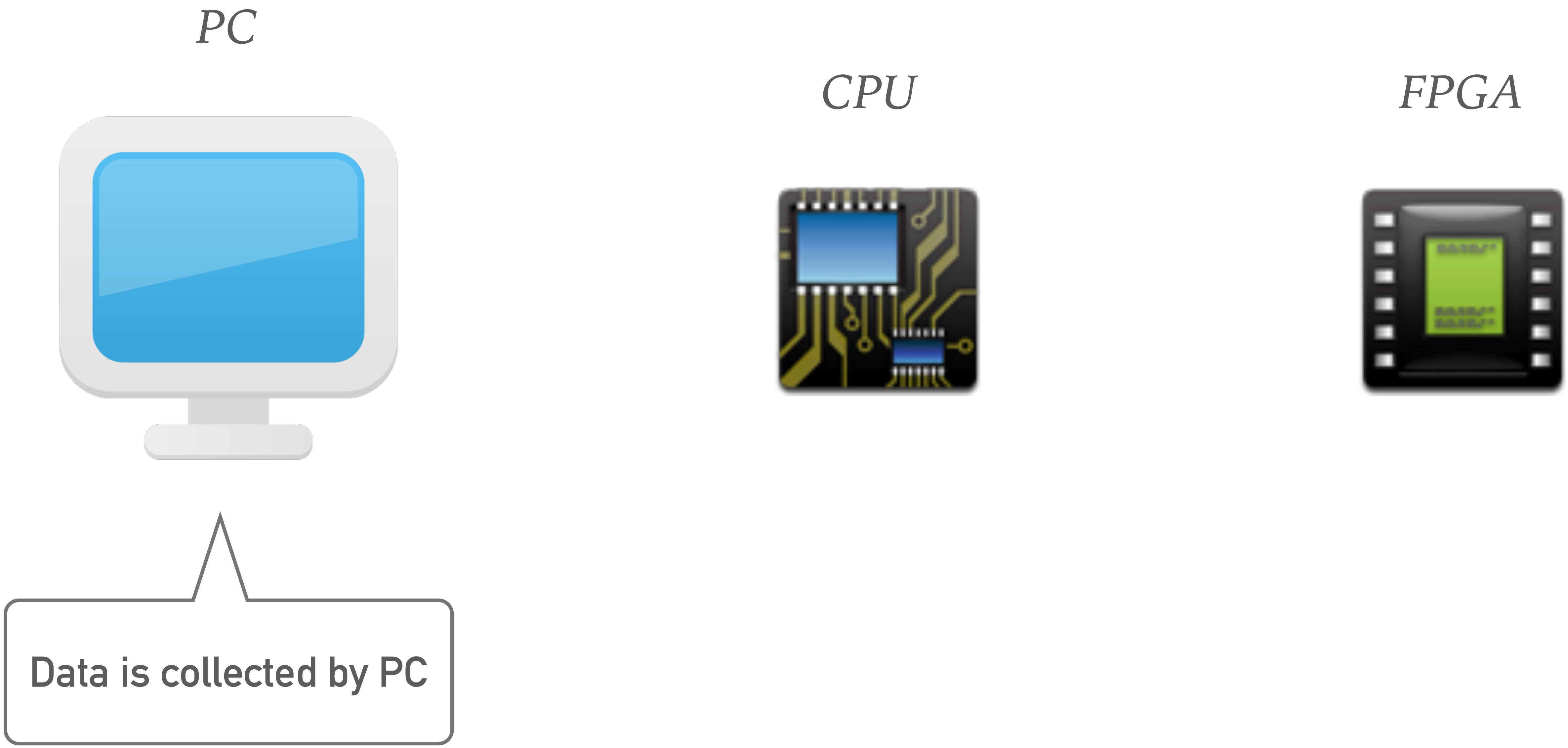


PC

CPU

FPGA

CPU COMMUNICATION MODULE – WORK FLOW



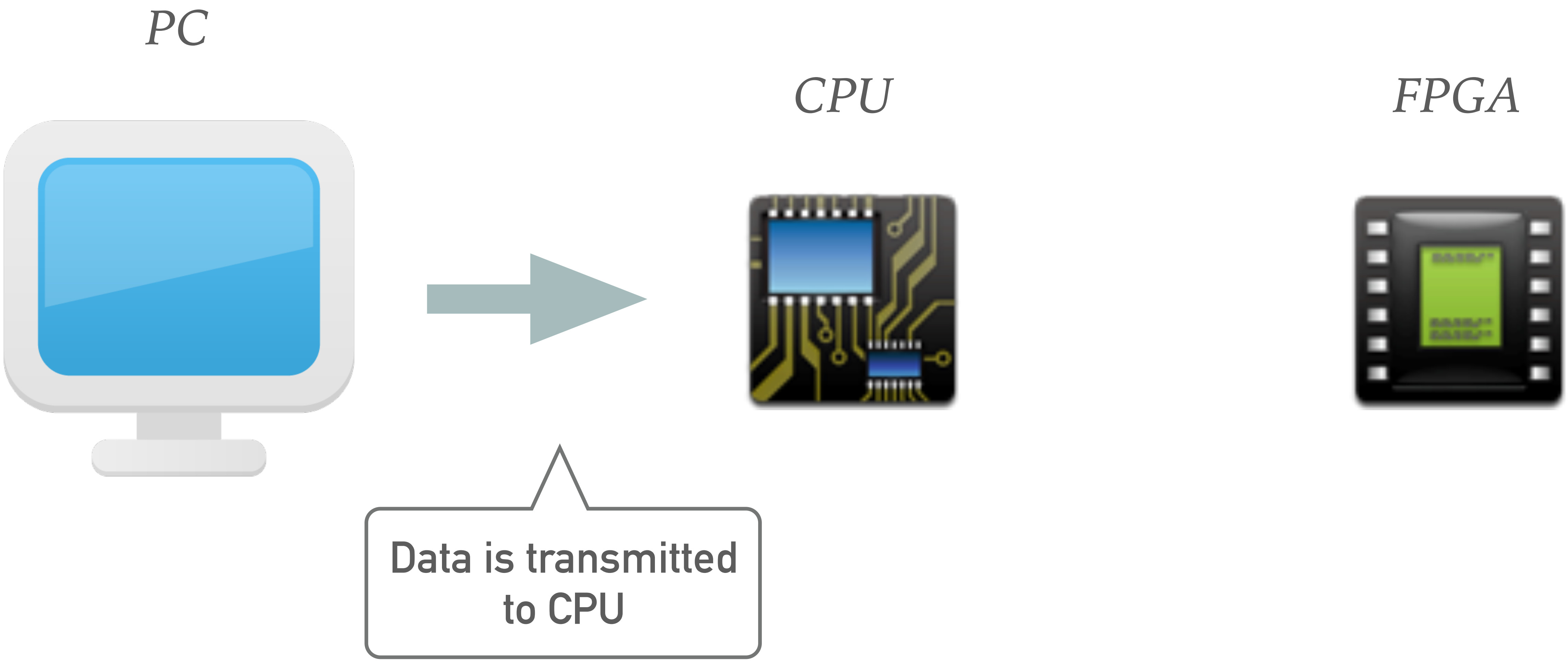
PC

CPU

FPGA

Data is collected by PC

CPU COMMUNICATION MODULE – WORK FLOW



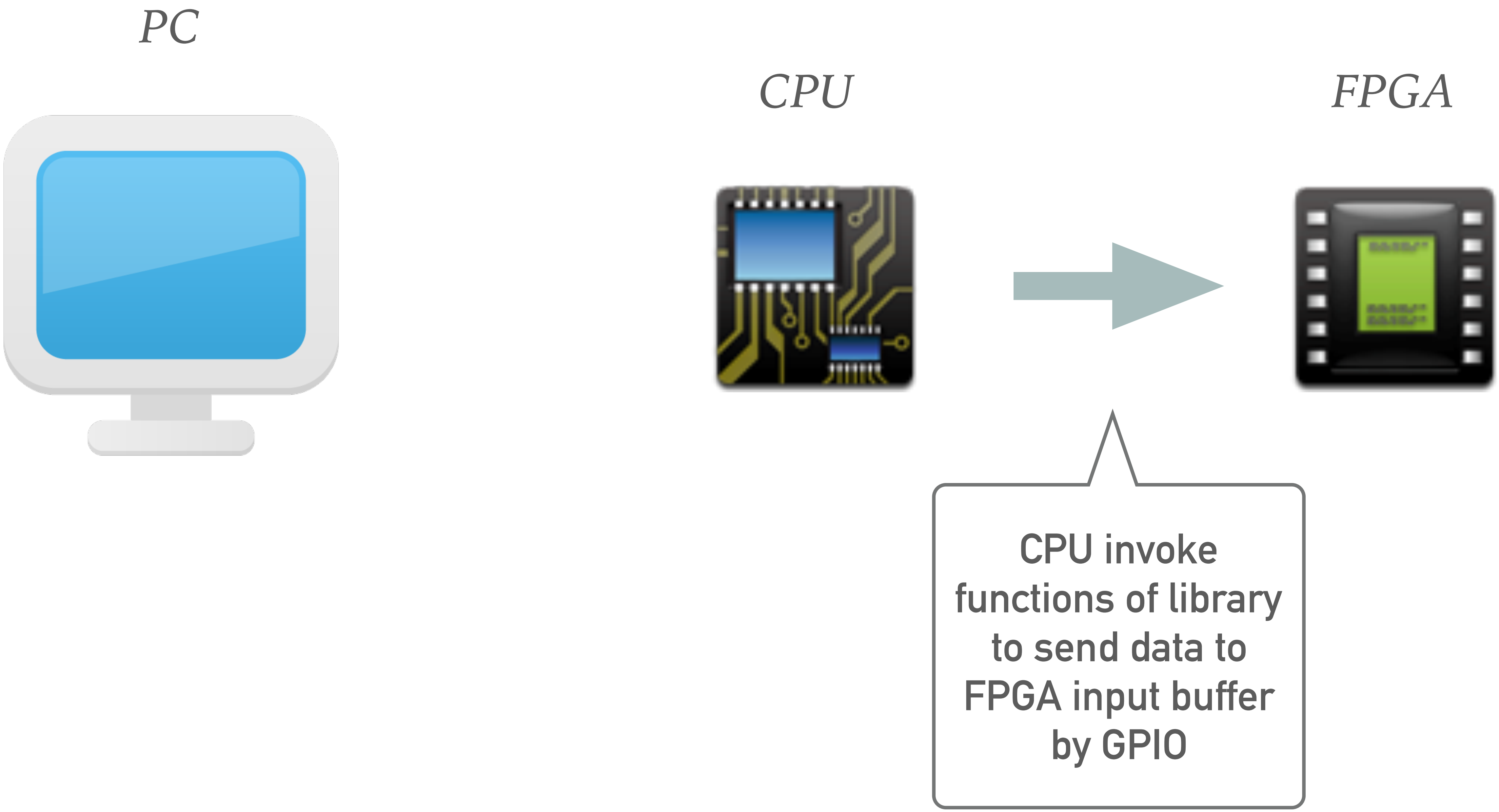
PC

CPU

FPGA

Data is transmitted to CPU

CPU COMMUNICATION MODULE – WORK FLOW



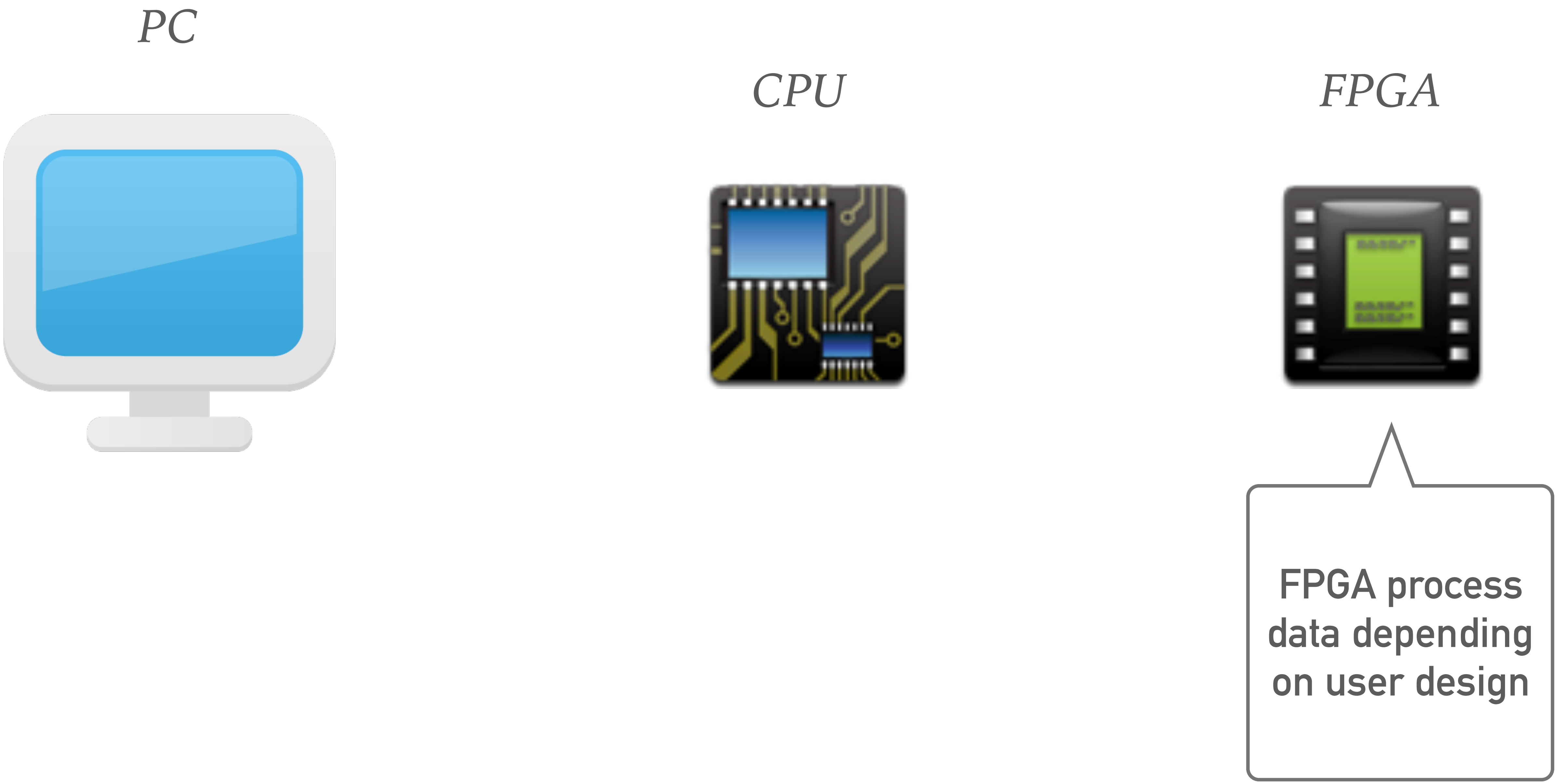
PC

CPU

FPGA

CPU invoke functions of library to send data to FPGA input buffer by GPIO

CPU COMMUNICATION MODULE – WORK FLOW



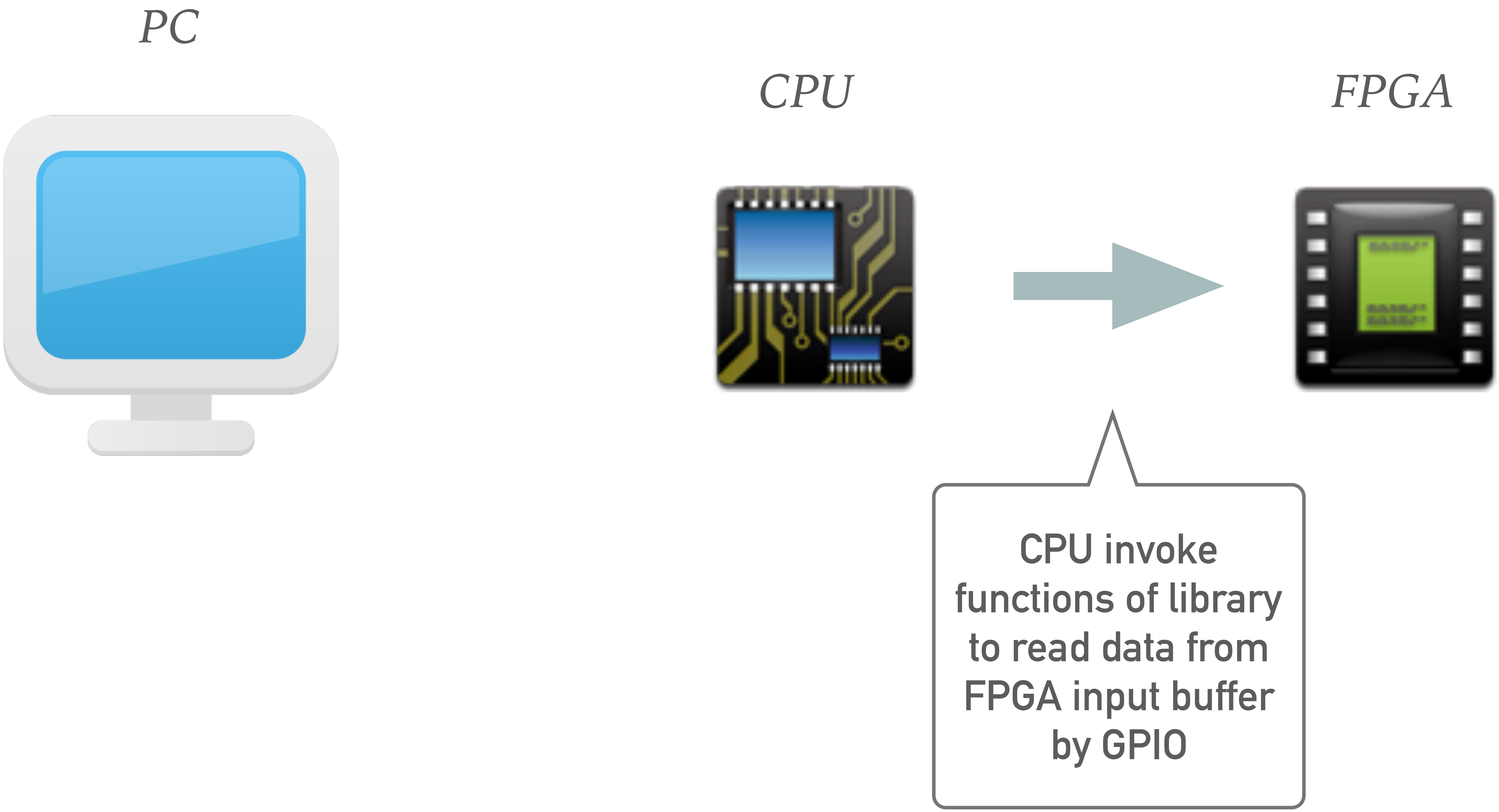
PC

CPU

FPGA

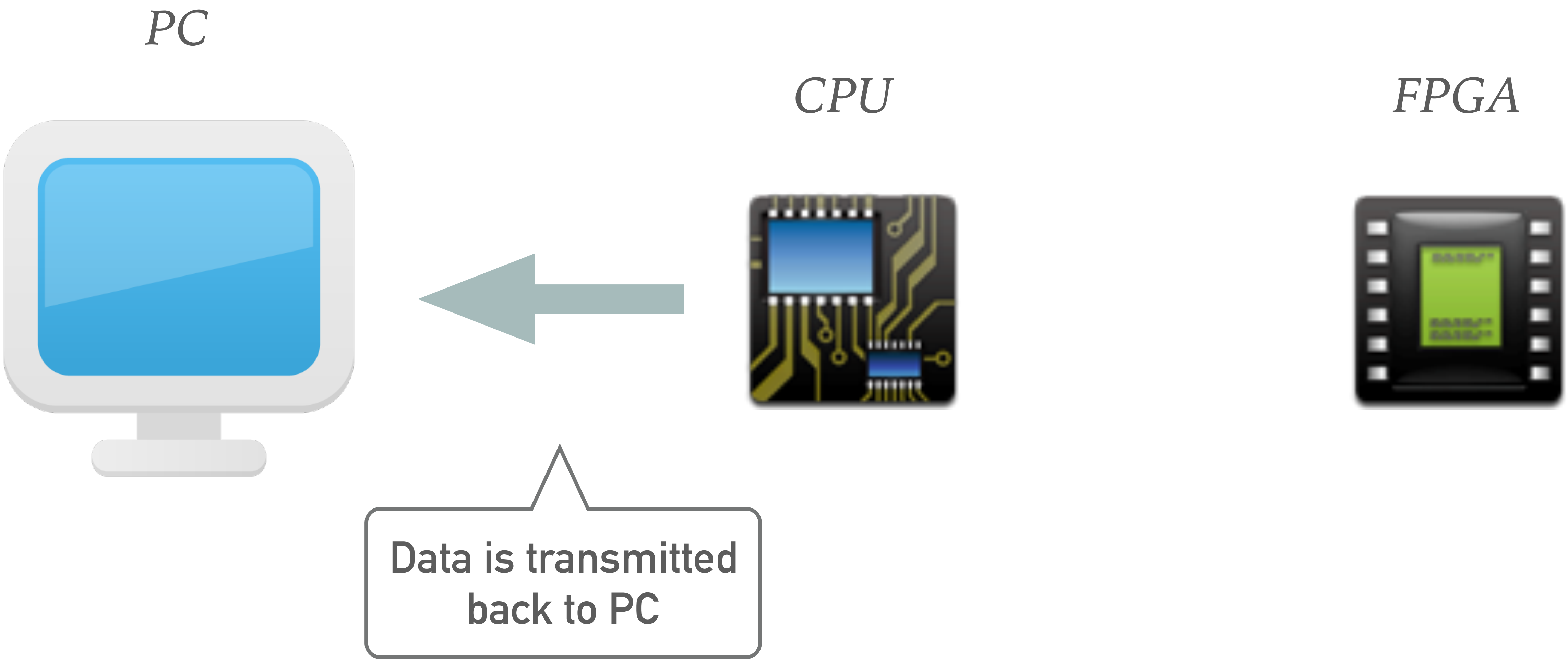
FPGA process data depending on user design

CPU COMMUNICATION MODULE – WORK FLOW



CPU invoke functions of library to read data from FPGA input buffer by GPIO

CPU COMMUNICATION MODULE – WORK FLOW



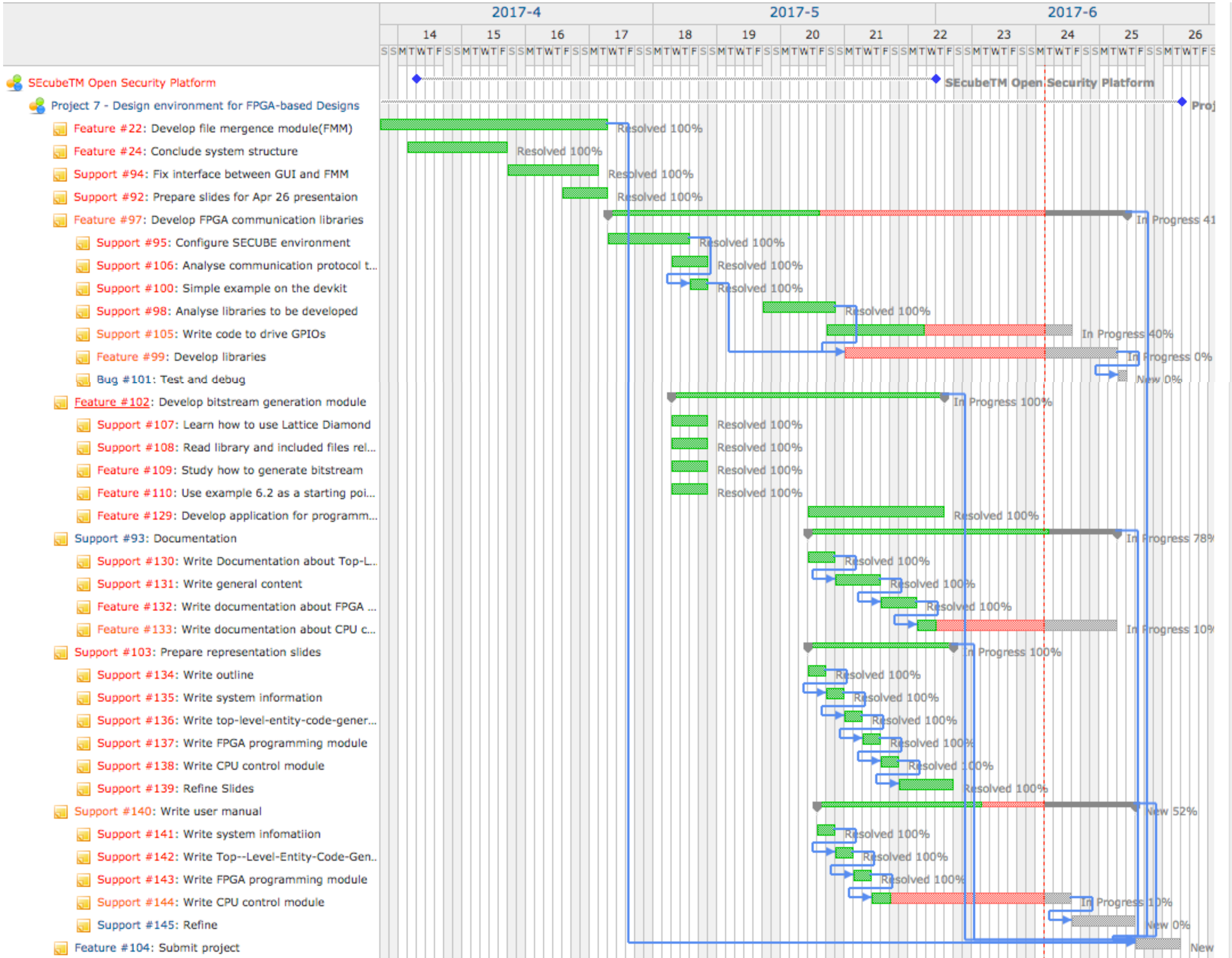
Data is transmitted back to PC

OUTLINE

- Introduction
- Top-Level code generator
- Bitstream generator
- CPU communication module
- Summary

SUMMARY

- Gnatt



SEcubeTM Open Security Platform
Project 7 - Design environment for FPGA-based Designs
Feature #22: Develop file mergence module(FMM)
Feature #24: Conclude system structure
Support #94: Fix interface between GUI and FMM
Support #92: Prepare slides for Apr 26 presentaion
Feature #97: Develop FPGA communication libraries
Support #95: Configure SECUBE environment
Support #106: Analyse communication protocol t...
Support #100: Simple example on the devkit
Support #98: Analyse libraries to be developed
Support #105: Write code to drive GPIOs
Feature #99: Develop libraries
Bug #101: Test and debug
Feature #102: Develop bitstream generation module
Support #107: Learn how to use Lattice Diamond
Support #108: Read library and included files rel...
Feature #109: Study how to generate bitstream
Feature #110: Use example 6.2 as a starting poi...
Feature #129: Develop application for programm...
Support #93: Documentation
Support #130: Write Documentation about Top-L...
Support #131: Write general content
Feature #132: Write documentation about FPGA ...
Feature #133: Write documentation about CPU c...
Support #103: Prepare representation slides
Support #134: Write outline
Support #135: Write system information
Support #136: Write top-level-entity-code-gener...
Support #137: Write FPGA programming module
Support #138: Write CPU control module
Support #139: Refine Slides
Support #140: Write user manual
Support #141: Write system infomatiion
Support #142: Write Top--Level-Entity-Code-Gen..
Support #143: Write FPGA programming module
Support #144: Write CPU control module
Support #145: Refine
Feature #104: Submit project

MILESTONES - DELIVERABLES

- Milestone 1: Finish top-level code generator

  Deliverables: Source files of top-level code generator

  Accomplished on **April 25**

- Milestone 2: Finish CPU communication module

  Deliverables: Source files of CPU communication module

- Milestone 3: Finish bitstream generator

  Deliverables: Source files of bitstream generator

  Accomplished on **June 1**

- Milestone 4: Submit project

  Deliverables: 1. Source files of entire project
  2. Documentation

Thank you!