Software Case Study: Autopilot

- The Autopilot application showcases some basic automated flight control functionalities for manned aircrafts and illustrate key concepts related to flight control systems such as:
  - Autopilot Flight Director System (AFDS),
  - Ground Collision Avoidance System (GCAS),
  - Waypoint navigation system (NAV),
  - Global Positioning System (GPS),
  - and Control & Display Unit (CDU)

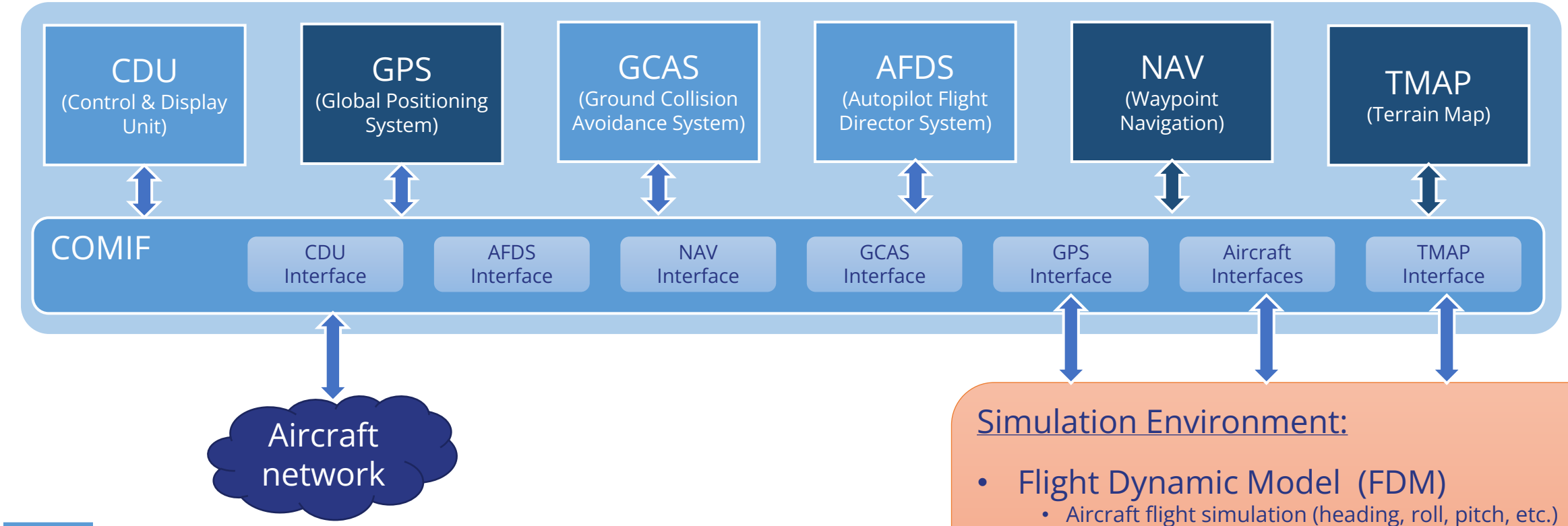
- Key Features:
  - Automated Control: The autopilot offers basic automated control of the aircraft's altitude, and velocity with GCAS functionality.
  - It is designed as a multi-components system with clear separation and defined communication between components.
  - For testing purpose, a simple Flight Dynamic Model (FDM) is provided (if available, an external FDM can be used as well).

- The CDU, AFDS and GCAS modules are written in Ada, the GPS, NAV and TMAP components are in C++.

- The application is supported by an RTOS with each component mapped to a different task.

- This software is open-source and released under MIT license.

Autopilot Application Components



Ada

C++

Simulation Environment:

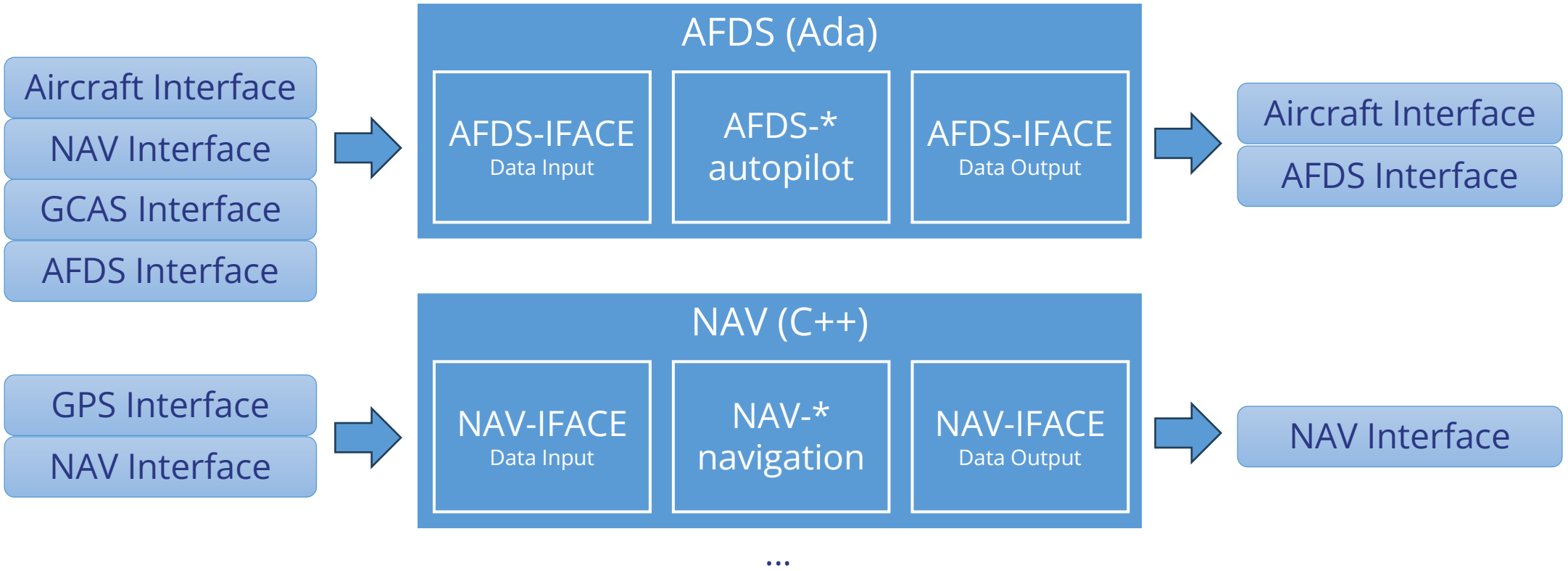
- Flight Dynamic Model (FDM)
  - Aircraft flight simulation (heading, roll, pitch, etc.)
  - GPS simulation (latitude, longitude, altitude)
- MAP server
  - Terrain elevation

Autopilot Components

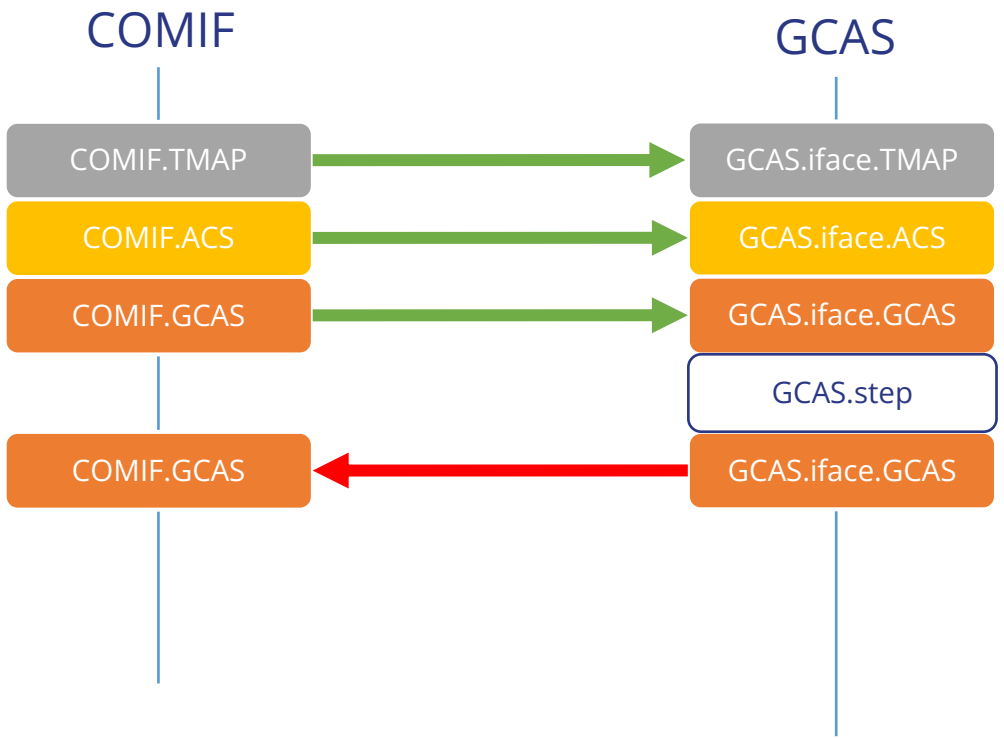
- Each component follows the same pattern
  - A component XYZ is made of two parts:
    - Data interface: XYZ.iface and XYZ.iface.*
    - The implementation: XYZ, XYZ.impl and XYZ.*
  - The XYZ package contains two entry points for the component:
    - ‘reset’: to reset the internal state (if any)
    - ‘step’: to execute one step of the component function
  - A step is made of 3 actions
    - Read data from the global interface (COMIF.* packages) to the local interface (XYZ.iface.*)
    - Perform the function(s) of the component
    - Write data back from the local interface (XYZ.iface.*) to the global interface (COMIF.*)

Components communication

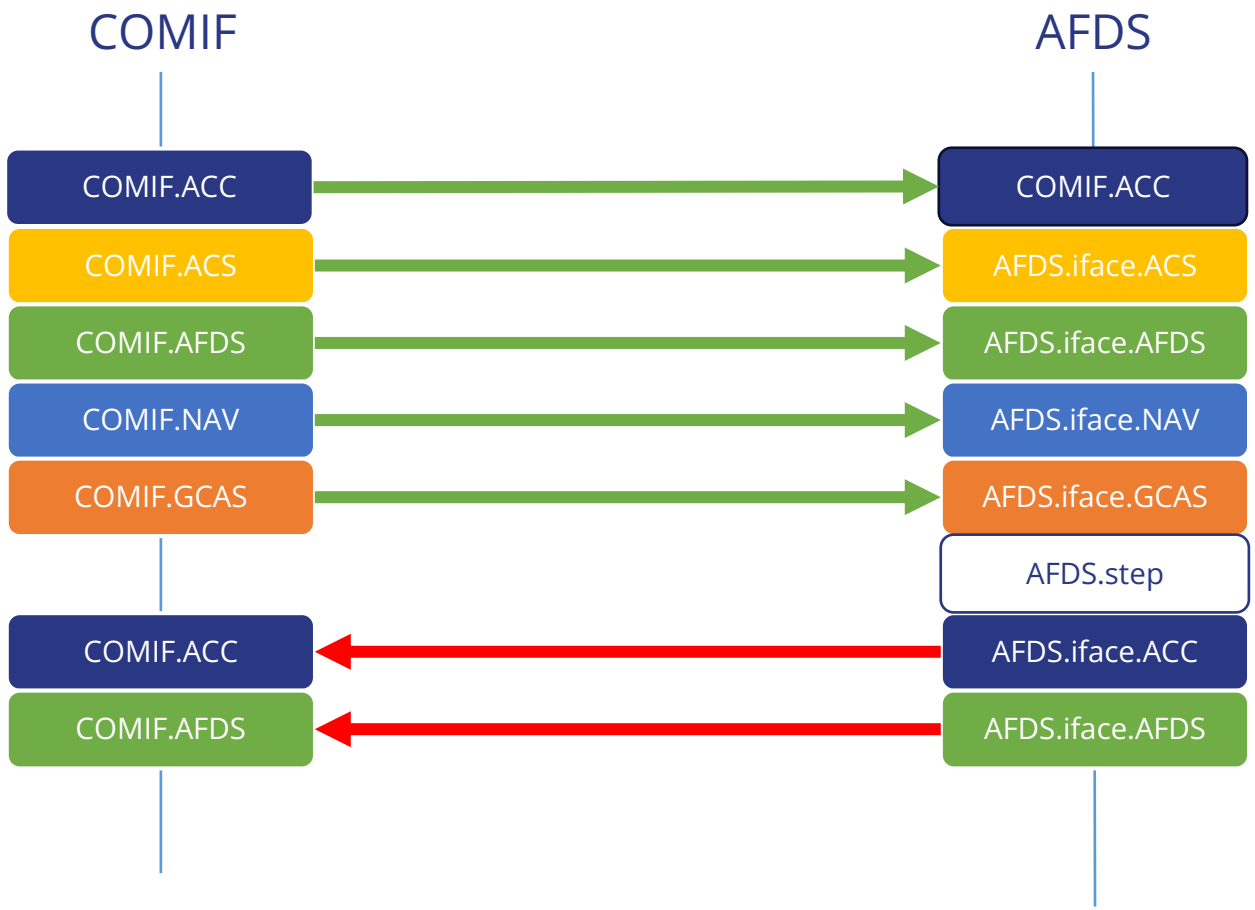
- Each component read and write data from/to the other components via COMIF
  - First, the required component data is read from COMIF in a local copy via the 'iface' package (e.g. AFDS.iface)
  - Then, the component function is executed (potentially modifying the local copy of another component data).
  - Finally, the required component data is written back to COMIF



Data flow example: GCAS



Data flow example: AFDS



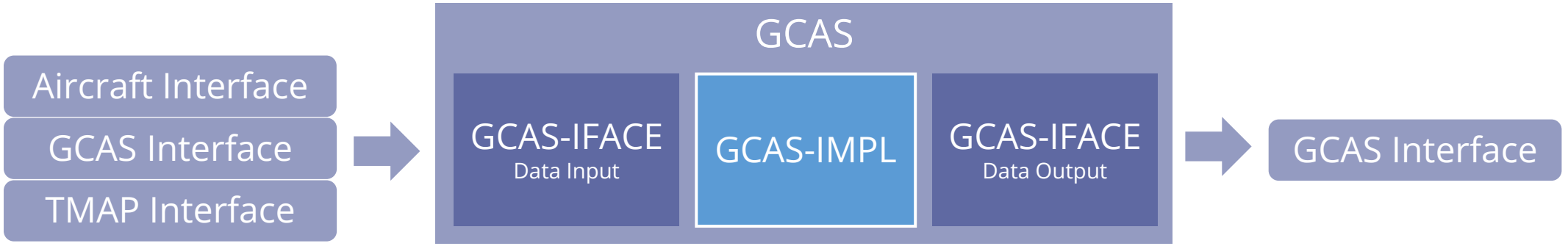
COMIF allowed data paths

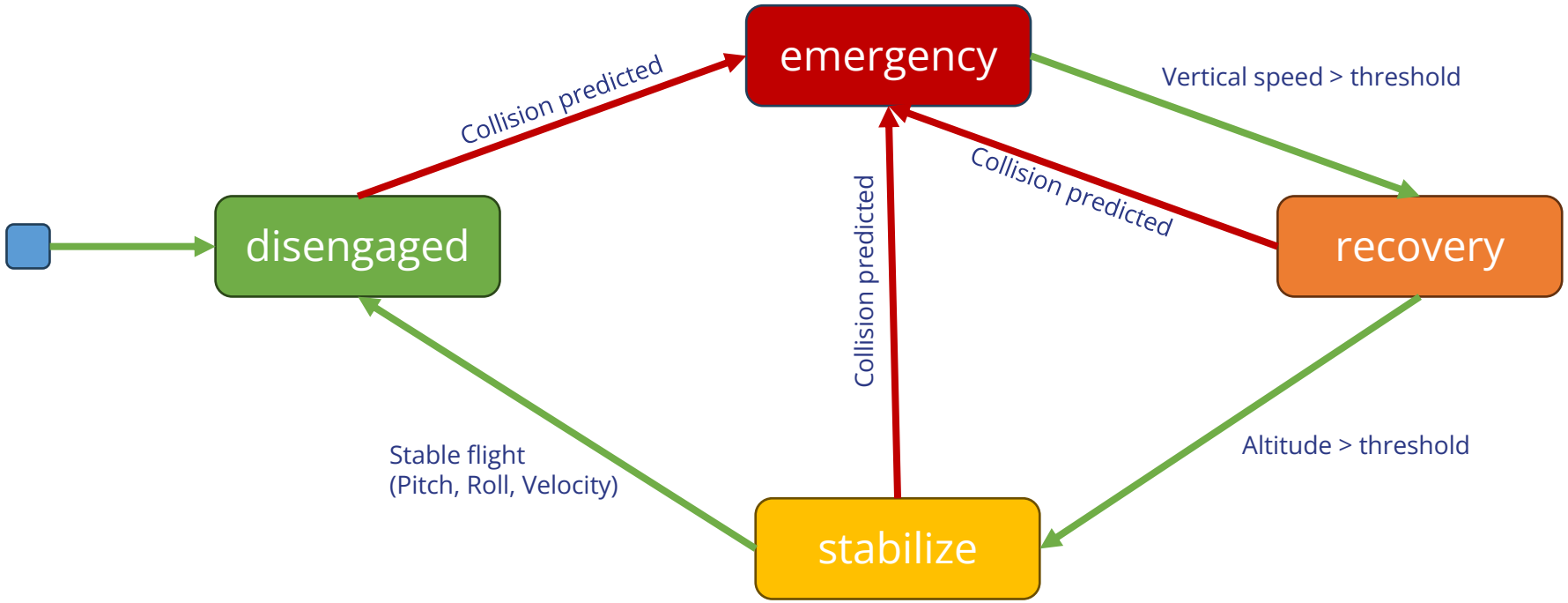
From\To	Aircraft status	Aircraft control	AFDS	GCAS	NAV	CDU	GPS	TMAP
AFDS	R	RW	RW	R	R	-	R	-
GCAS	R	-	-	RW	-	-	R	R
NAV	R	-	-	-	RW	-	R	-
CDU	-	-	RW	RW	RW	RW	-	-
GPS	-	-	-	-	-	-	RW	-
TMAP	-	-	-	-	-	-	R	-
FDM	RW	R	-	-	-	-	RW	RW
TEST	RW	RW	RW	RW	RW	RW	RW	RW

GCAS

- The GCAS component detect potential ground collision and set the GCAS state accordingly (see next slide)

- The GCAS state is checked by the AFDS component and the AFDS.GCAS function of this component implements the required manoeuvre.


Aircraft Interface
GCAS Interface
TMAP Interface
GCAS
GCAS-IFACE
Data Input
GCAS-IMPL
GCAS-IFACE
Data Output
GCAS Interface


emergency
Collision predicted
Vertical speed > threshold
disengaged
Collision predicted
Collision predicted
recovery
Stable flight
(Pitch, Roll, Velocity)
Altitude > threshold
stabilize

NAV

- This component is implemented in C++
- The NAV component navigate to the next waypoint.
- When a waypoint is reached, the next waypoint in the list is selected as the active waypoint.
- Navigation data (desired heading, altitude, velocity) are written to the IFACE.NAV interface for the AFDS component to read.

GPS

- This component is implemented in C++
- Receive latitude/longitude/altitude from equipment (via COMIF)
- Calculate and publish (to COMIF.GPS)
  - Ground speed
  - Vertical speed

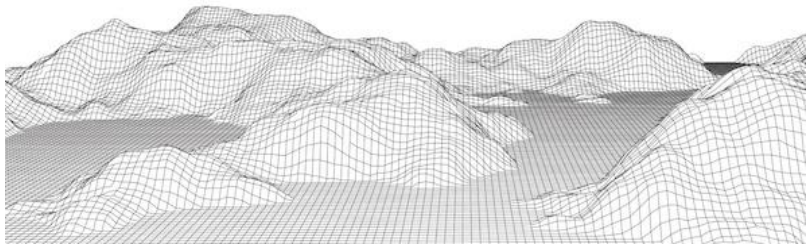
CDU

- The CDU component can:
  - Enable/Disable GCAS, AFDS and NAV
  - NAV:
    - Clear the NAV waypoint list
    - Add a waypoint list
  - AFDS
    - Set the desired navigation target

- The CDU component receives
  - The state of GCAS, AFDS and NAV (enabled/disabled)

TMAP

- This component is implemented in C++
- The TMAP component:
  - Maintains a local map of ground elevation
    - Map is 20km x 20km, centred on the aircraft position with a resolution of 100m for position and 10m for elevation.
  - Communicates with the TMAP server (part of simulated environment) to request new parts of the map when the aircraft position changes.
  - Updates the current ground elevation at the aircraft position.
  - Computes the elevation profile for the next 5 seconds of flight on the current heading (used by GCAS)

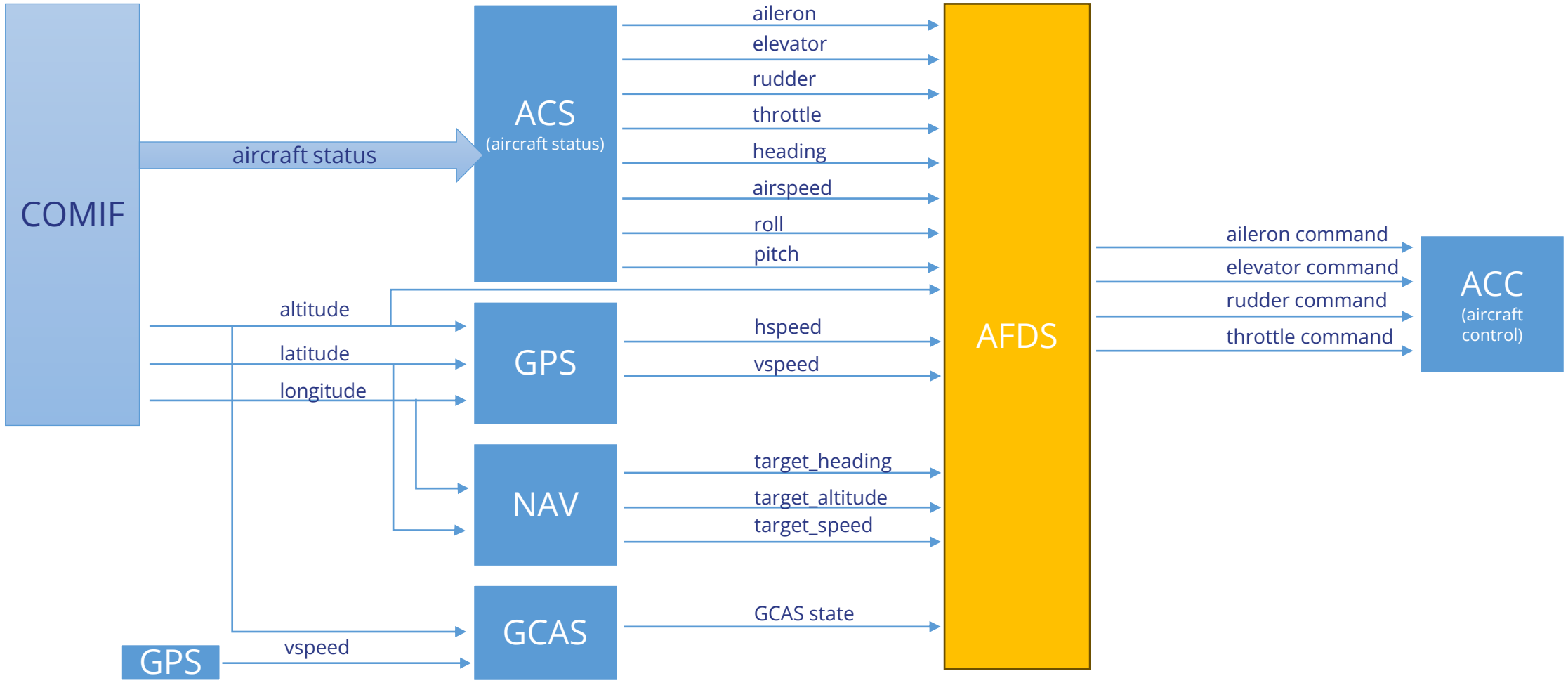


AFDS

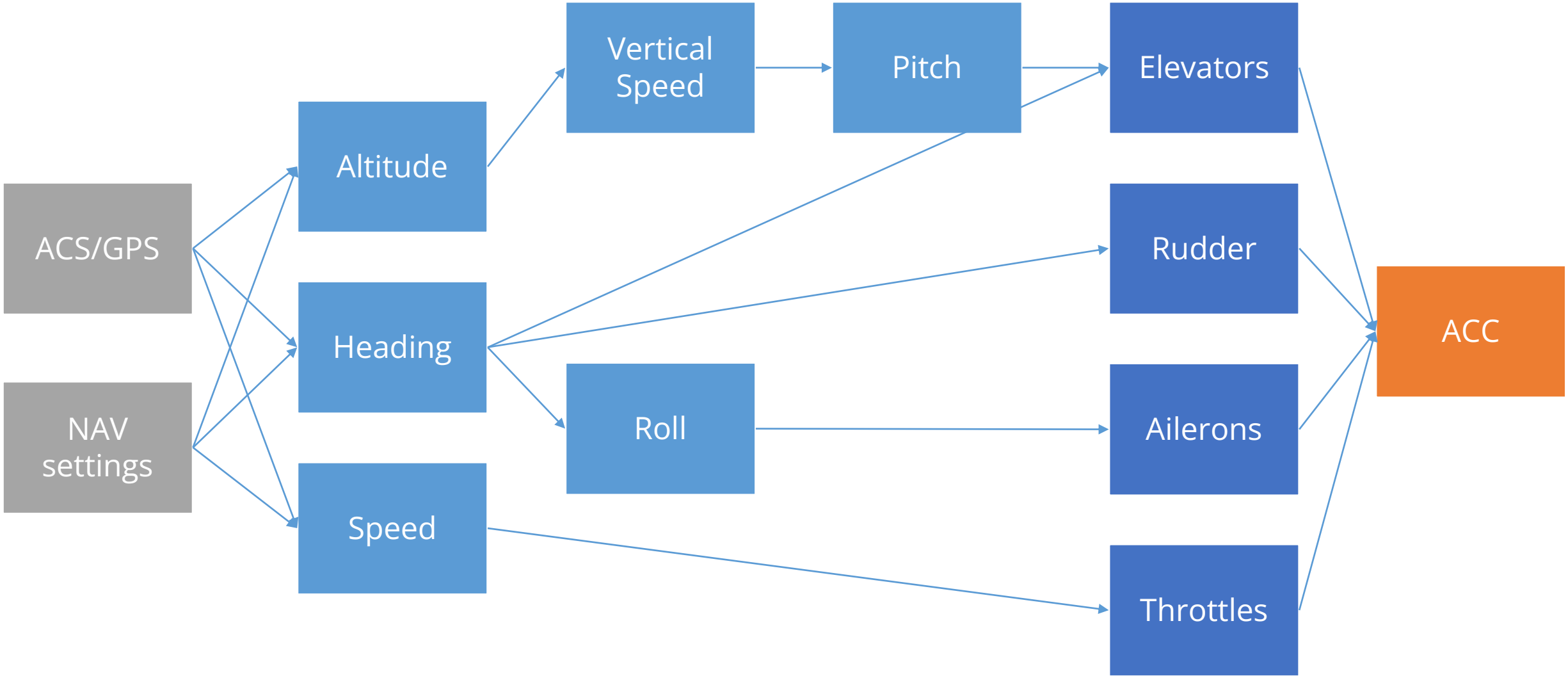
- The AFDS read the current and desired heading, altitude and velocity set by the AFDS, NAV or CDU components and calculate the require commands for the aileron, elevator, rudder and throttle.

- Note: AFDS.GCAS can override the NAV input to force an emergency collision avoidance manoeuvre.

AFDS


COMIF
aircraft status
ACS
(aircraft status)
aileron
elevator
rudder
throttle
heading
airspeed
roll
pitch
altitude
latitude
longitude
GPS
hspeed
vspeed
NAV
target_heading
target_altitude
target_speed
GCAS
GCAS state
GPS
vspeed
AFDS
aileron command
elevator command
rudder command
throttle command
ACC
(aircraft control)

AFDS-* (Autopilot)



AFDS-GCAS override