Build a car simulator with

YAQS v1

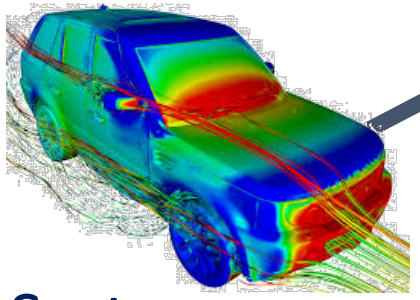
Version 1 : very basic…

Start with only 6 modules

Sound environment

I/O

Instruments panel



Systems

Navigation

3D Visual System

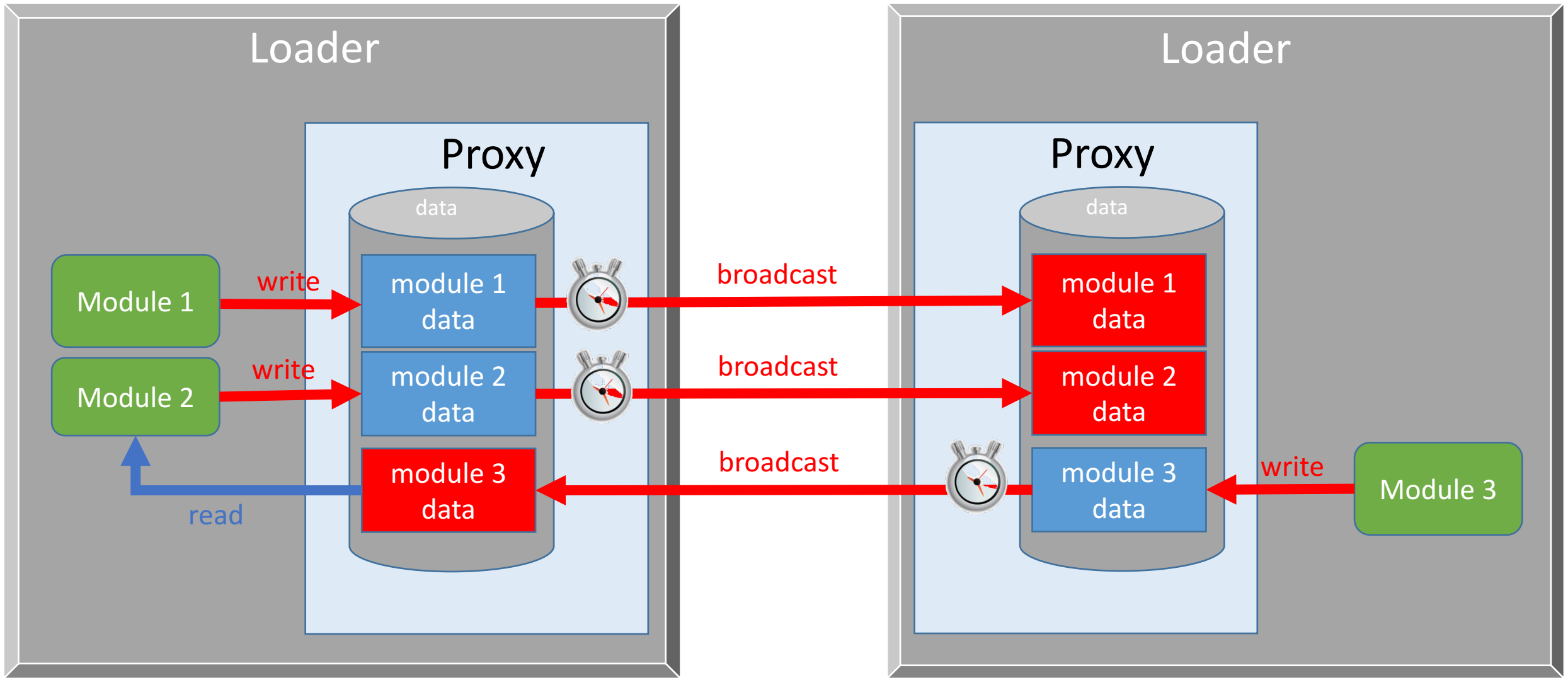
Simple distributed architecture

- A commun data dictionnary is defined
- Every item of the dictionnary is written by only one YAQS module
- A YAQS loader is an application having :
  - at least one module
  - a YAQS proxy
- A YAQS proxy is in charge of providing the commun data to every modules.
- Data are synchronize along YAQS proxy using UDP broadcast

Commun data dictionnary

#MsgID	Module	Nom	Description	Valeur
1	IO	IO_dperc_WheelPos	The wheel position	-100 (gauche)…100(droite)
1	IO	IO_perc_ThrottlePos	The throttle pedal position	0 (released)…100 (pressed)
2	SYSTEMS	SYST_degmn_latitude	The pilot eye position	lat, long, alt
2	SYSTEMS	SYST_degmn_longitude		
2	SYSTEMS	SYST_deg_heading	heading	heading in deg
2	SYSTEMS	SYST_kmh_Speed	The car speed	km/h
2	SYSTEMS	SYST_rpm_EngineRPM	The engine RPM	rpm

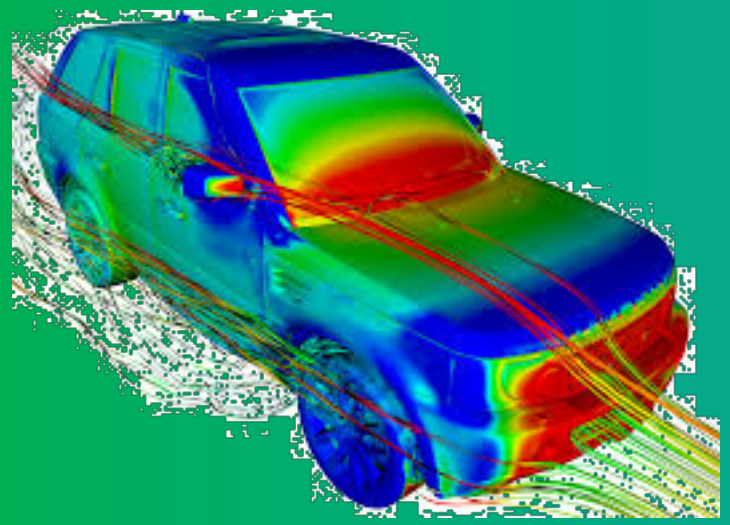
overview


Loader
Proxy
data
Module 1
write
module 1 data
broadcast
module 1 data
Module 2
write
module 2 data
broadcast
module 2 data
read
module 3 data
broadcast
module 3 data
write
Module 3
Loader
Proxy
data

10

- Move the steering wheel
- Move the pedal

Systems



- The pedal position (from I/O) increase the RPM
- The RPM drives the speed with an inertia
- The wheel position updates the heading
- The position is updated with speed and the heading

Sound

- An engine sound is played :
  - engine RPM is used to determine the frequency and volume.

Navigation

- A map with the car position

Instrument panel

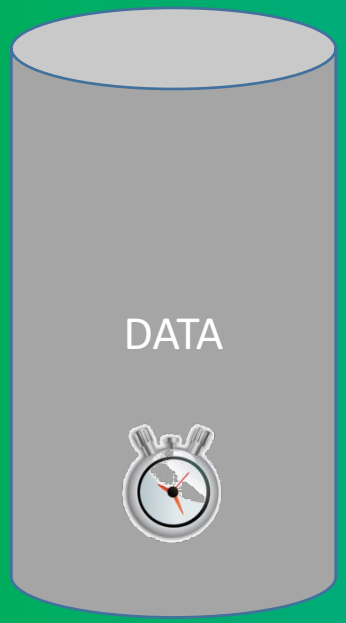
- Display speed and engine RPM

Visual system

- a flat ground textured with nantes map

What’s next ?

Architecture



- Make Proxy discoverable (Hello)
- Replace the static commun dictionnary with a dynamic connection of modules i/o
- Support several synchronization mode :
  - UDP / TCP / Bluetooth / IPC…
  - make proxies discoverable and dynamically define the best sync strategy
- Create a tool to define the modules connection

IO

- Handle more switches
- Support additionals way of inputs (sensors for moving wheels, joystick…)

Misc

- Connect 2 cars