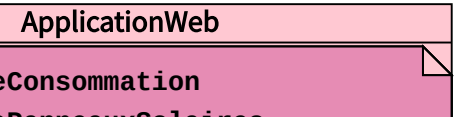
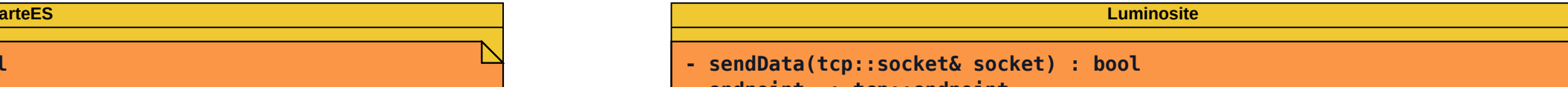
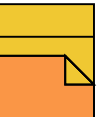
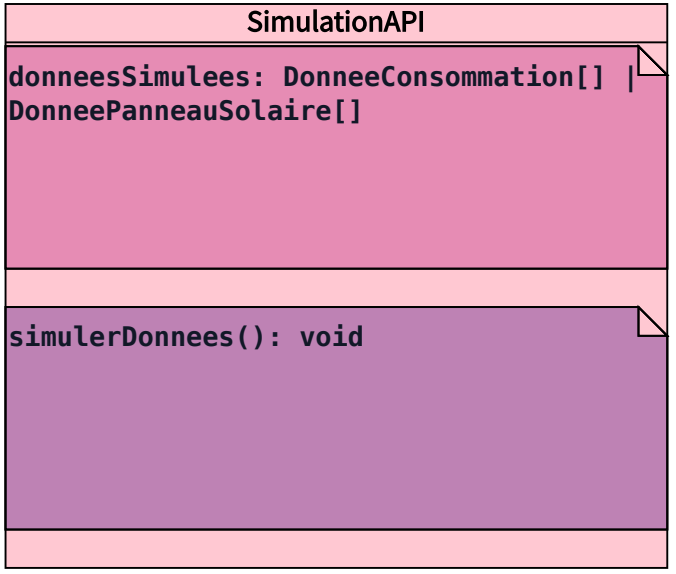
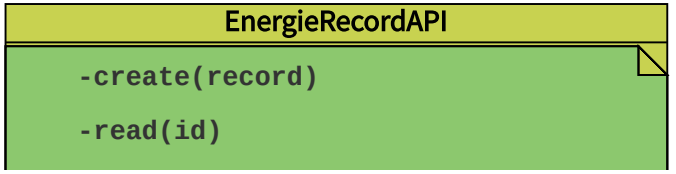
graphique

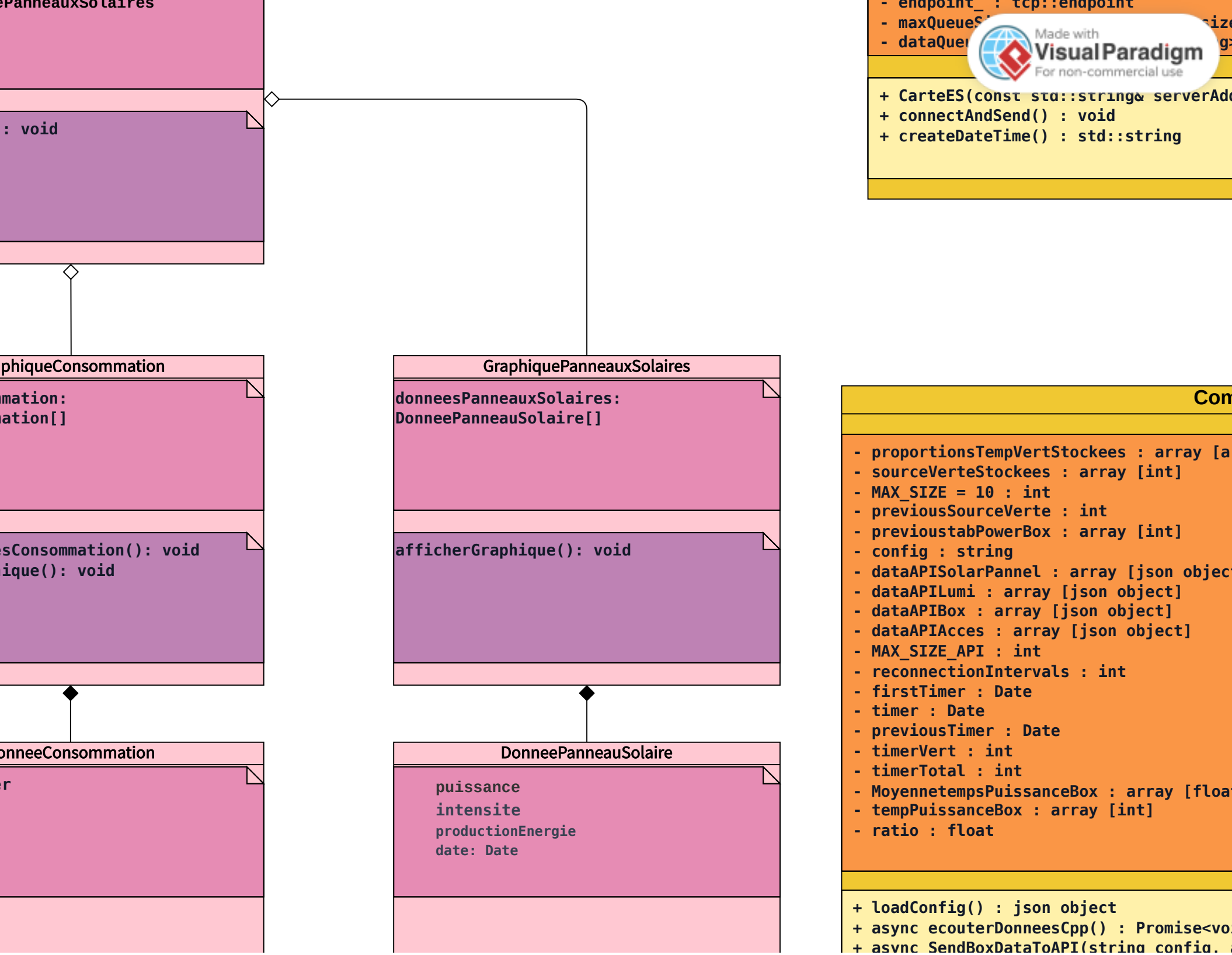




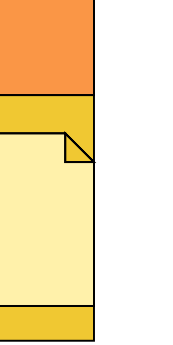




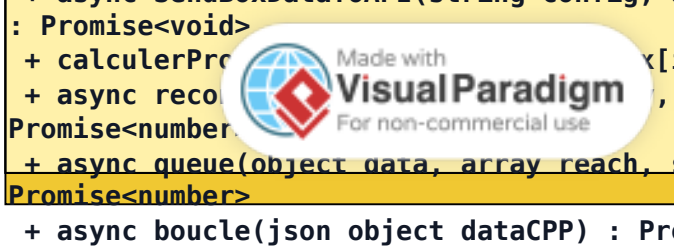



ePanneauxSolaires
: void
+ CarteES(const std::string& serverAdd
+ connectAndSend() : void
+ createDateTime() : std::string
Made with VisualParadigm For non-commercial use
phiqueConsommation
mation:
ation[]
sConsommation(): void
ique(): void
GraphiquePanneauxSolaires
donneesPanneauxSolaires:
DonneePanneauSolaire[]
afficherGraphique(): void
Com
- proportionsTempVertStockees : array [a
- sourceVerteStockees : array [int]
- MAX_SIZE = 10 : int
- previousSourceVerte : int
- previoustabPowerBox : array [int]
- config : string
- dataAPISolarPannel : array [json objec
- dataAPILumi : array [json object]
- dataAPIBox : array [json object]
- dataAPIAcces : array [json object]
- MAX_SIZE_API : int
- reconnectionIntervals : int
- firstTimer : Date
- timer : Date
- previousTimer : Date
- timerVert : int
- timerTotal : int
- MoyennetempsPuissanceBox : array [floa
- tempPuissanceBox : array [int]
- ratio : float
+ loadConfig() : json object
+ async ecouterDonneesCpp() : Promise<vo
onneeConsommation
r
DonneePanneauSolaire
puissance
intensite
productionEnergie
date: Date


e_t
>
dress, int serverPort)
- endpoint_ : tcp::endpoint
- maxQueueSize = 3 : static const size_t
- dataQueue_ : std::deque<std::string>
+ LUMINOSITE(const std::string& serverAddress, int serverPort)
+ connectAndSend() : void
+ createDateTime() : std::string
Made with
VisualParadigm
For non-commercial use
npteurs
rray[float]]
t]
t]
id>
array data[], string dataKey)
Capteur_luminosite
- m_carte : int
- numcarte : int
- sendData(tcp::socket& socket) : bool
- serverAddress_ : std::string
- serverPort_ : int
- maxQueueSize = 3 : static const size_t
- dataQueue_ : std::deque<std::string>
- endpoint_ : sockaddr_in
- resistance = 44.6 : double
+ capteur_luminosite(const std::string& serverAddress, int serverPort, int car
PCI_9111DG)
+ ~capteur_luminosite()
+ lire_tension(double& tension, int canal = 0, int gamme = AD_B_10_V) : void
+ lire_tension(int canal = 0, int gamme = AD_B_10_V) : double
+ lire_tension_AI_ReadChannel(int canal = 0, int gamme = AD_B_10_V) : unsigned
+ getnumcarte() : int
+ setnumcarte() : void
+ connectAndSend() : void
+ createDateTime() : std::string
+ getresistance() : double



```
-update(record)
-delete(id)
```


: Promise<void>
+ calculerPro
+ async reco
Promise<number
+ async queue(object data, array reach,
Promise<number>
+ async boucle(json object dataCPP) : Pr
Made with
VisualParadigm
For non-commercial use

```
array data[]) string dataKey)

int]) : json object
 array reach, string dataKey) :

string configKey, string dataKey) :
```

```
omise<void>
```