



Installation et personnalisation du serveur de données NAVISU

Ref : TV050714_TU_SM

Rédacteur : Serge Morvan

Présentation

Ce tutoriel fait suite à celui ci : Architecture et fonctionnalités du serveur de données . Il a pour objet l'explication de la démarche complète d'installation, de paramétrisation, voire de la modification, du serveur de données capteurs. Ce serveur fait l'acquisition des données, assure leur multiplexage ainsi que la diffusion de ces données au format **xml**. Il est nécessaire d'installer ce serveur, dans le cas d'une application distribuée où l'acquisition et la visualisation ne se fait pas complètement ou uniquement au sein de NaVisu.

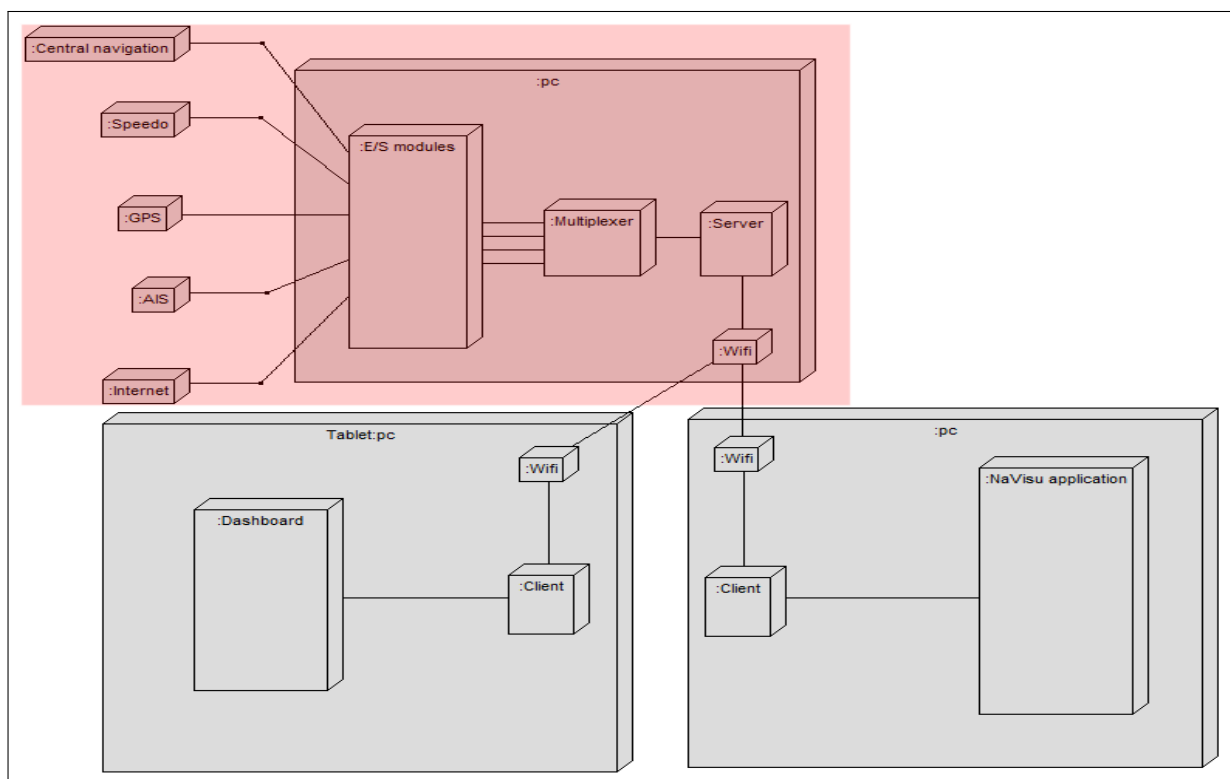


FIGURE 1 – *Déploiement distribué*



Le projet

Le projet NAVISU est divisé en sous-projets, chaque sous-projet possède une architecture type composants. La plupart des sous-projets sont développés sous forme d'API, ayant le minimum de dépendance avec les autres sous-projets. C'est le cas de l'API **navisu-server**, qui ne dépend que du module **navisu-domain** : description des modèles d'objets utilisés. Il est donc simple de présenter le serveur sous forme d'un projet indépendant.

Téléchargement

<https://github.com/terre-virtuelle/Navisu-server.git>

Paramétrisation

l'API NaVisu-server ne fourni pas d'IHM, pour la paramétrisation, celle ci se fait à l'aide du fichier de propriétés : **properties/server.properties**

```
# NmeaServer properties file

#File name for tests without serial comm
fileName = data/nmea/gps.txt

# Web server parameters
hostName = localhost
port = 8080
queueSize = 5

# Serial parameters
# [portNumber - 1] for Linux
portNumber = 5
# COM for Windows OS, /dev/ttyS for Linux
portName = COM5
baudRate = 4800
dataBits = 8
stopBits = 1
parity = 0
```

Modifier les paramètres en fonction de votre configuration. Attention à la dénomination différente des ports sous Windows et sous Linux.

La variable **port** correspondant au numéro du port de communication avec les clients, attention de n'avoir pas déjà des applications utilisant ce port.

Lancement

A partir du fichier jar : `java -jar NaVisuServer.jar`

Développement

La classe de test est : `bzh.terrevirtuelle.navisu.server.app.ServerMain`

```
ComponentManager componentManager = ComponentManager.componentManager;
// deploy components
LOGGER.log(Level.INFO, "\n Start", componentManager.startApplication(
    DataServerImpl.class
));
DataServerServices nmeaServerServices =
    componentManager.getComponentService(DataServerServices.class);

// Test avec choix des parametres de comm
// nmeaServerServices.init("localhost", 8080);
// nmeaServerServices.openSerialPort("COM4", 4800, 8, 1, 0);
// nmeaServerServices.openSerialPort("COM5", 4800, 8, 1, 0);

// Test avec les parametres de comm dans properties/nmea.properties
nmeaServerServices.init();
nmeaServerServices.openSerialPort();
// nmeaServerServices.openFile();
```

La première partie du code est relative à la gestion des composants. Ensuite deux séries d'exemples la première série fixe dans le code les paramètres d'acquisition et de diffusion. La deuxième série utilise les valeurs par défaut indiquées dans le fichier de propriétés. L'initialisation doit être unique, par contre on peut ouvrir autant d'entrées que l'on veut.