

Projet Données Réparties
Sciences du Numérique – Semestre 7

Architecture de principe

Enzo PETIT Nam VU

29 novembre 2021

Plan de travail initial

1. **CentralizedLinda** sans gestion des callbacks
 - a) Sauvegarde des tuples en mémoire (**write**) — *Nam*
 - b) Méthodes non bloquantes (**tryTake**, **tryRead**, **takeAll**, **readAll**) — *Enzo*
 - c) Méthodes bloquantes (**take**, **read**) — *Nam*
2. **CentralizedLinda** avec callbacks (**eventRegister**) — *Enzo*
3. Classes client/mono-serveur sans callbacks — *Enzo*
4. Classes client/mono-serveur avec callbacks — *Nam*

L'écriture de tests se fera au fur et à mesure de manière croisée par chacun d'entre nous (e.g. Nam écrira les tests sur les méthodes non bloquantes qu'Enzo a implémenté).

Architecture de principe

Classes envisagées

Pour la version mémoire partagée à priori rien de plus que la classe **CentralizedLinda**.

Pour la version client/mono-serveur on aura les classes **LindaClient**, **LindaServer** (interface du serveur) et **LindaServerImpl** (implémentation du serveur).

Difficultés identifiées

Quelle structure de donnée Java pour sauvegarder les tuples reçus ?

Organisation des tests

Tests unitaires sous JUnit (ou sans si ça s'avère trop compliqué/long à mettre en place).