

Livraison COO

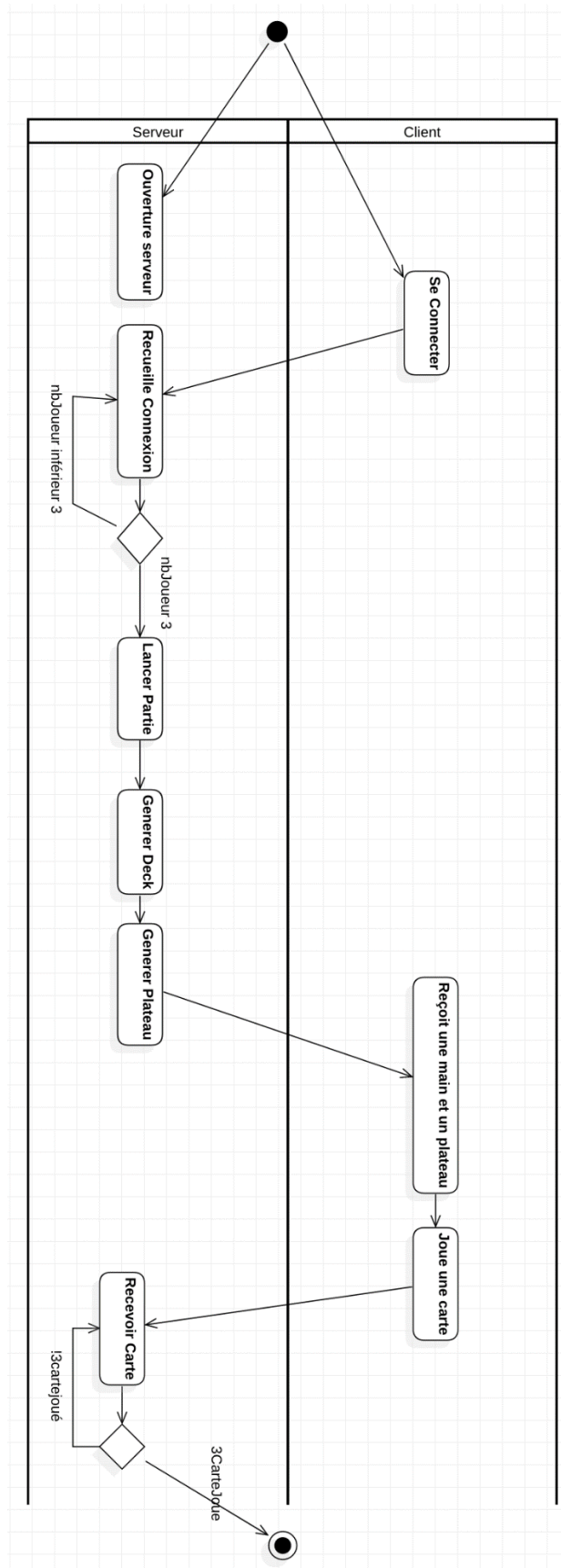
Groupe L

Samuel CUSTODIO CAVACO, Filipe DOUTEL SILVA, Hugo HABRIKH, Alexis
CHARLET, Sörel NADAUD

Table des matières

Point de vue général de l'architecture.....	2
Client.....	3
Diagramme de use case	3
Diagramme de classes.....	3
Diagrammes de séquence	4
Se connecter	4
Jouer une carte	4
Serveur	5
Diagramme de use case	5
Diagramme de classes.....	6
Diagrammes de séquence	6
Gérer deck	6
Gérer plateau.....	7
Interactions entre les clients et le serveur	7
Conclusion	7

Point de vue général de l'architecture



Client

Diagramme de use case

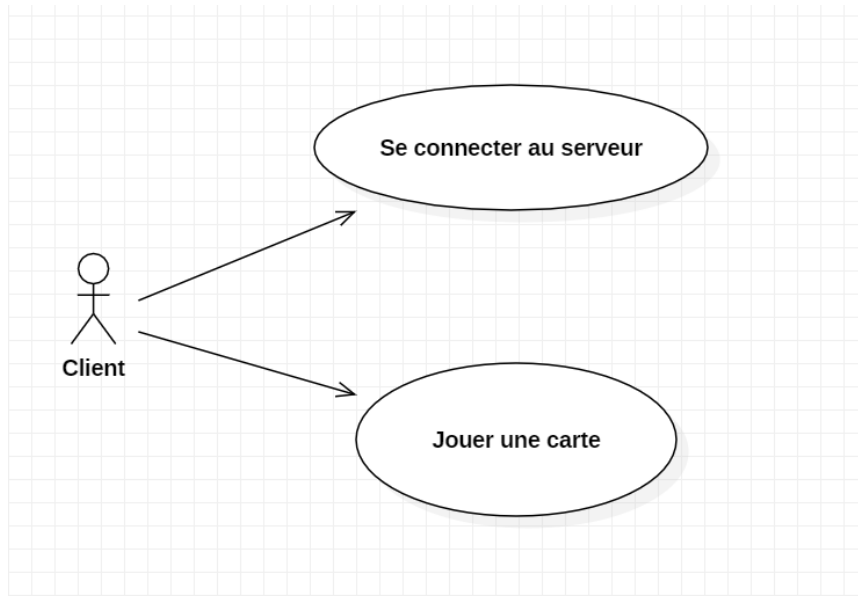
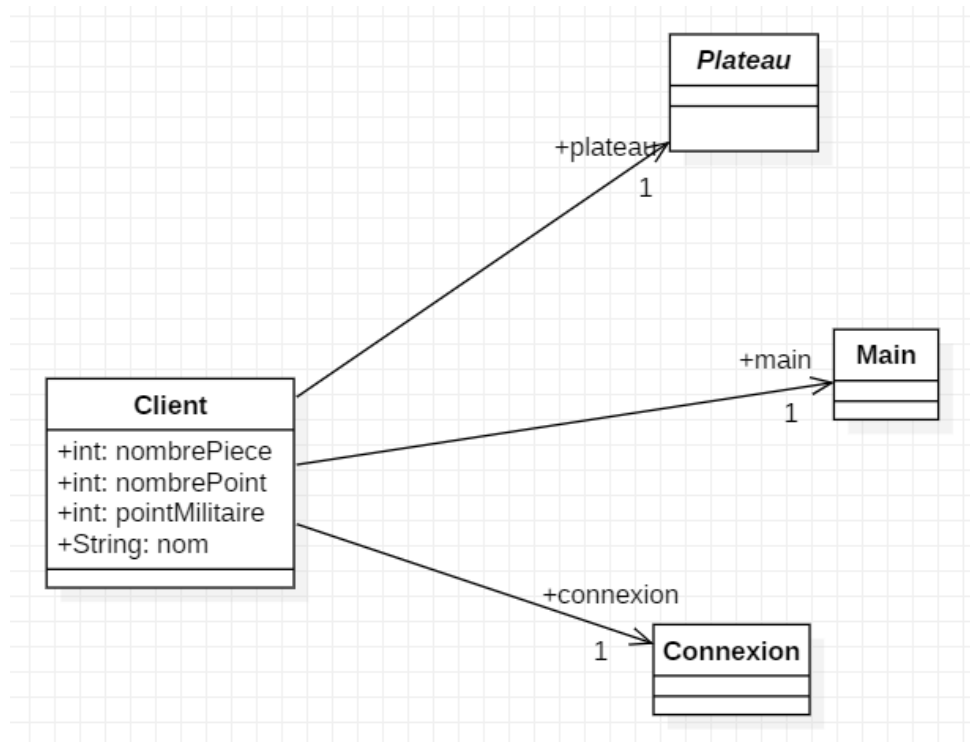


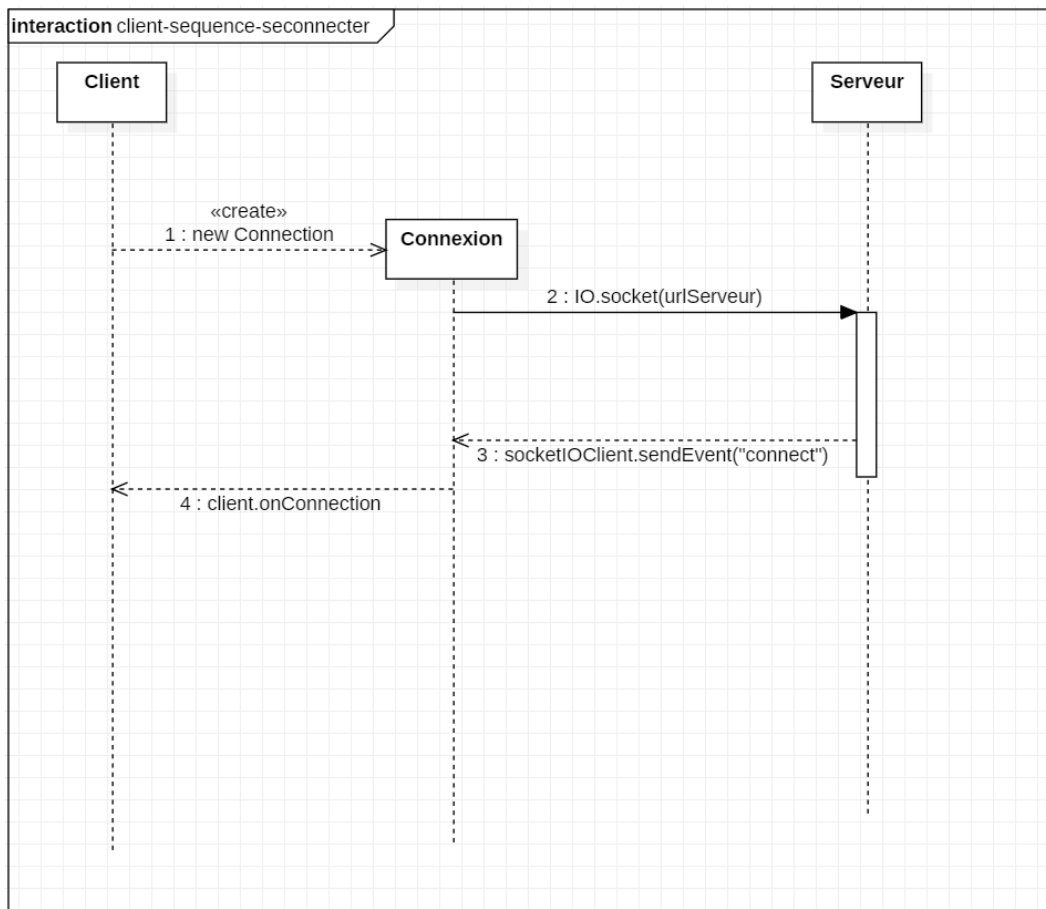
Diagramme de classes



Le client a un plateau, une main et une connexion au travers de laquelle il interagit avec le serveur.

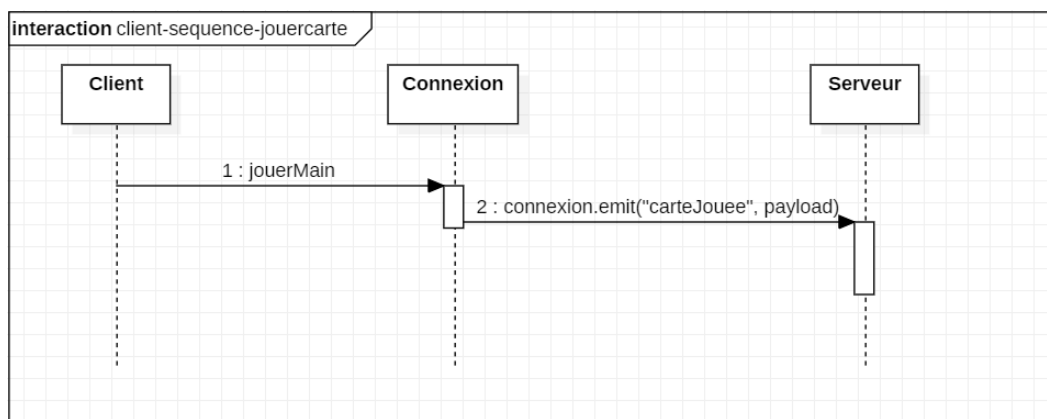
Diagrammes de séquence

Se connecter



A son initialisation, le client crée une connexion au travers de laquelle il va interagir avec le serveur pour se connecter. Le serveur répond au client avec un évènement « connect ». Le client possède une méthode « onConnection » qui est appelée lors de la réception de la réponse du serveur.

Jouer une carte



Le client possède une méthode « jouerMain » qui va, au travers de la connexion du client, émettre un évènement « carteJouee » en direction du serveur.

Serveur

Diagramme de use case

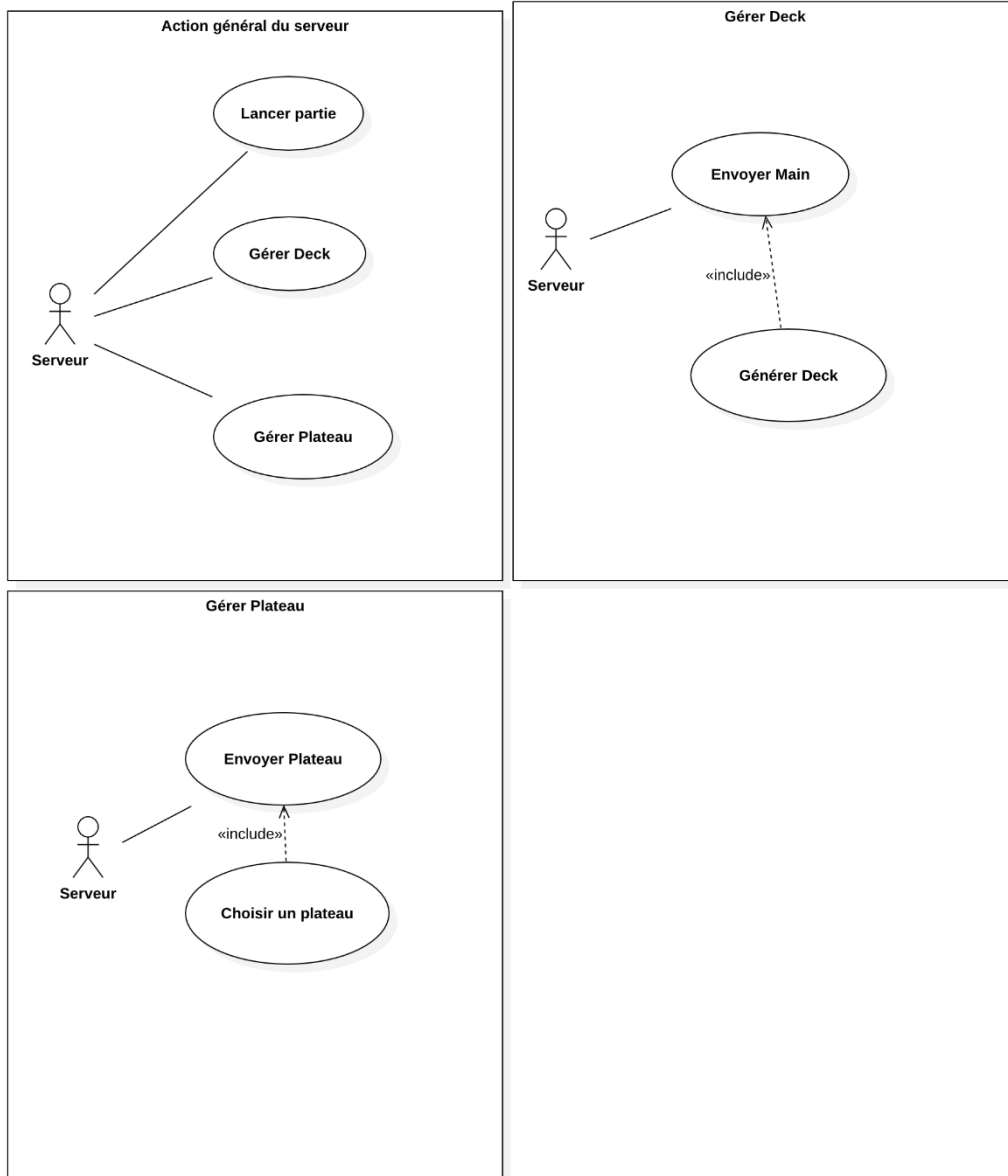
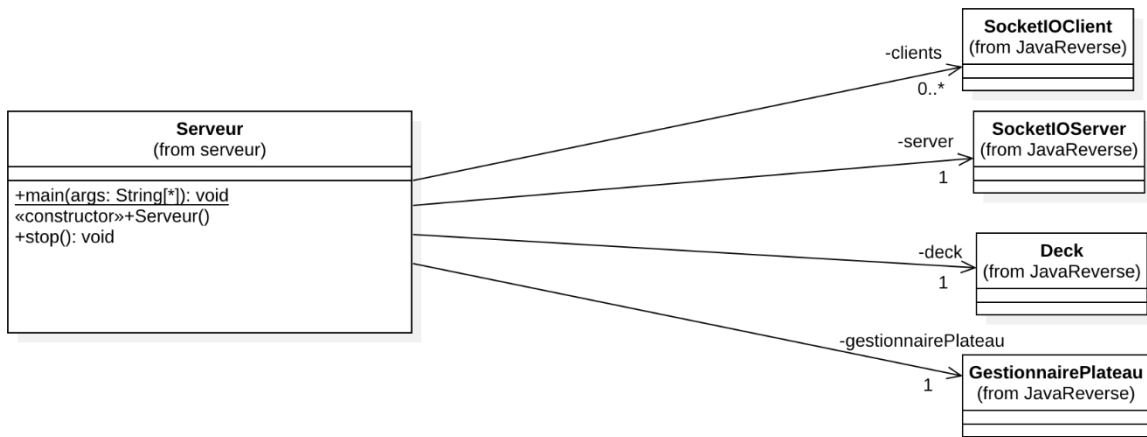
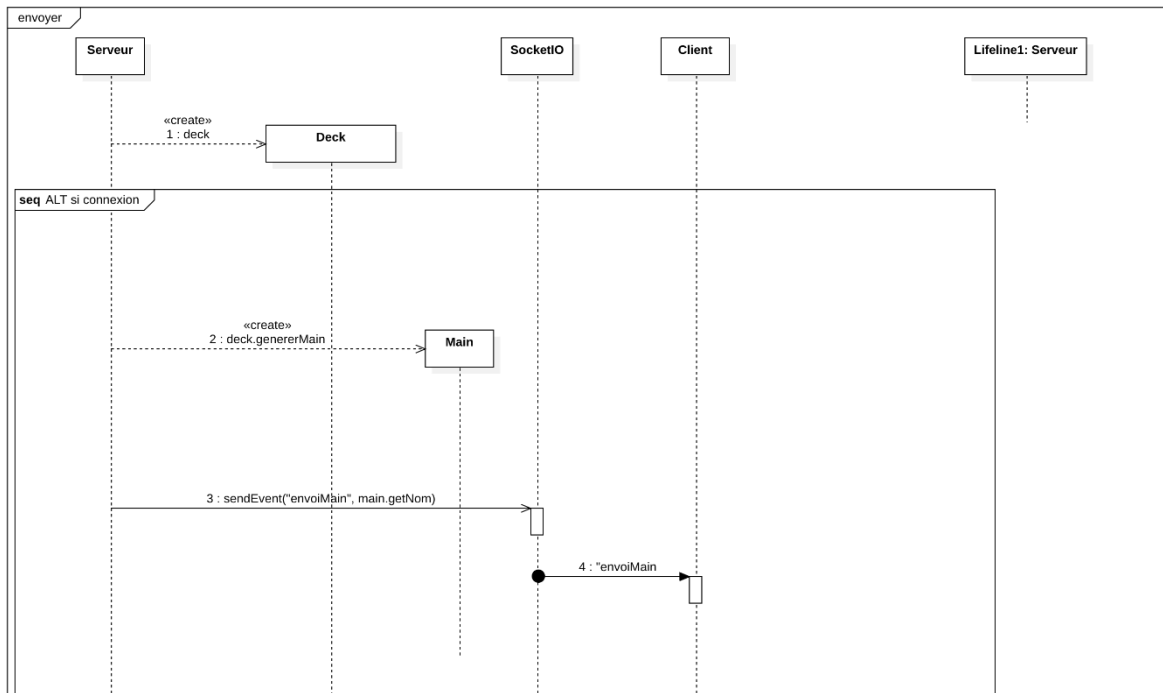


Diagramme de classes



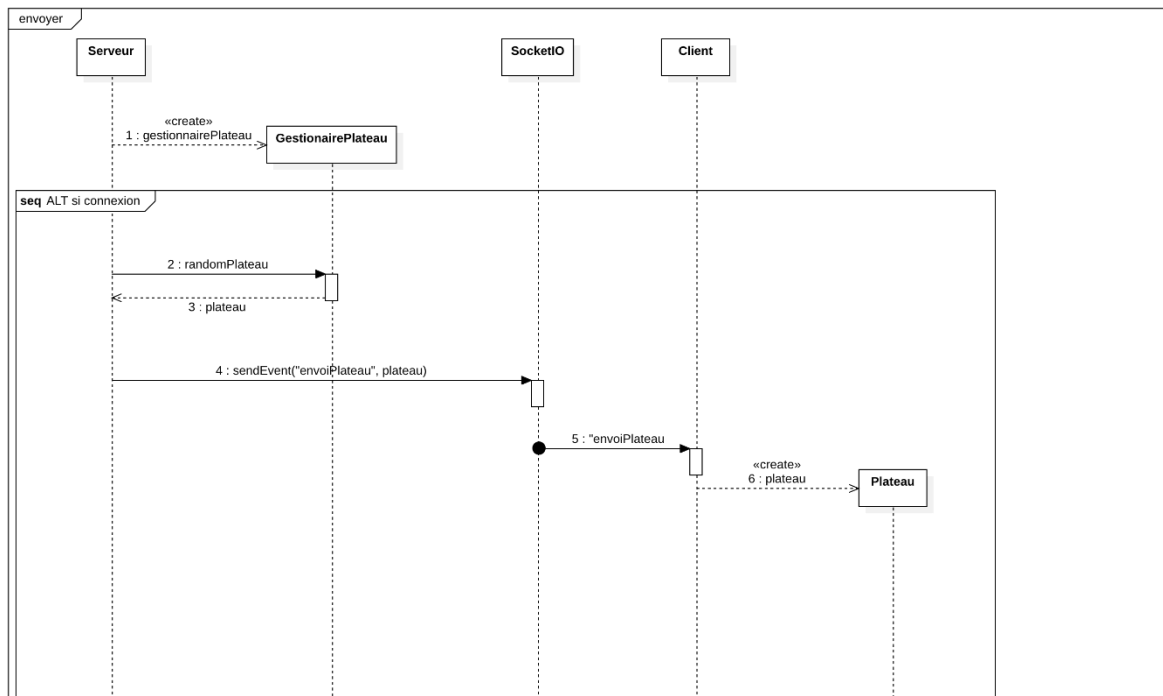
Diagrammes de séquence

Gérer deck



A son initialisation, le serveur crée un deck. Lorsqu'une connexion est détectée, une main est générée à partir du deck créé précédemment. Cette main est ensuite envoyée au client au travers d'une socket.

Gérer plateau



A son initialisation, le serveur crée un gestionnaire de plateaux. Lorsqu'une connexion est détectée, un plateau aléatoire est sélectionné à partir du gestionnaire de plateau créé précédemment. Ce plateau est ensuite envoyé au client au travers d'une socket.

Interactions entre les clients et le serveur

Tout d'abord l'échange de données entre le client et le serveur se fait via une socket. La communication repose essentiellement sur la transmission et la réception de String.

Lorsque la partie commence c'est le serveur qui va créer un deck et une main pour chaque joueur. Cette main ainsi qu'un plateau sera envoyé à chaque joueur.

En ce qui concerne le client, il va se connecter au serveur grâce à l'url de ce dernier couplé à un port. La principale communication se fera lorsque que chaque joueur aura une interaction avec les cartes ou le plateau.

Conclusion

Notre conception est opérationnelle, le point fort et que nous avons approfondi notre réflexion en début de projet pour prévoir toutes les règles et les fonctions de notre projet. De ce fait, nous avons deux classes distinctes carte et plateau, qui sont complètes et modulables.

Cependant, même si les données de nos classes sont bien structurées, nous avons eu un problème lors de la transmission de ces données. Le problème étant que nous ne pouvons pas directement communiquer une carte. De ce fait, nous avons dû passer par le système d'introspection et ainsi recréer chaque élément côté serveur à partir d'un String.