



27/04/2013

# Protocoles & Organisation

Puissance 5 – IPI S2 2013

Nathalie Au – Alexandre Métraud  
Jérémy Beaume – Alexandre Lewin  
ENSIIE – GROUPE G (4.2)

# Protocoles d'échanges de messages

Dans l'optique de pouvoir diviser le projet en modules répartis entre les différents programmeurs, les protocoles d'échanges de messages et les diagrammes de séquences ont été conçu en groupe préalablement. Ainsi, les programmeurs sauront exactement ce qu'ils recevront et devront envoyer aux autres modules du projet.

Nonobstant, ces protocoles sont susceptibles d'être légèrement modifiés entre la version alpha et la version beta des différents modules : des optimisations réseaux peuvent se révéler aux intervenants au cours du développement.

Afin de minimiser la taille des messages transmis sur le réseau, nous choisissons de précéder nos données par un code à un caractère unique par partie plutôt que des mots clés.

## Table des matières

Préliminaires .....	2
Types des variables .....	2
Carte des serveurs physiques .....	2
Phase d'initialisation .....	3
Protocoles .....	3
Diagramme de séquence .....	4
Phase de jeu .....	5
Protocoles .....	5
Diagramme de séquence .....	6
Gestion de projet .....	7
Responsabilités et travail en cascade .....	7

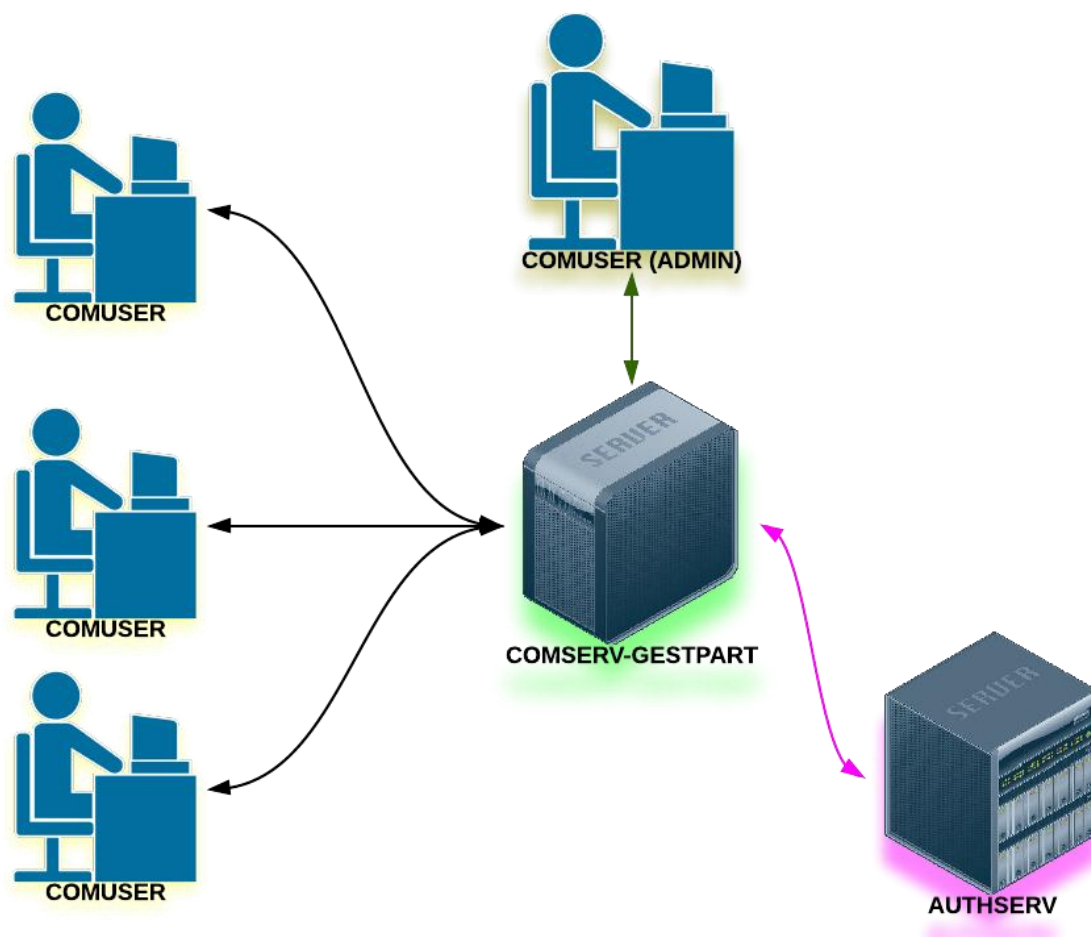
## Préliminaires

### Types des variables

Variable	Type	Exemple
Pseudo, joueur1, ..., joueurN, winner	CHAR[16]	« Alexandre », « Jérémy », « Nathalie »
Mot de passe	CHAR[16]	« pantoufle42 »
Largeur, Hauteur, Id	INT[2]	10
Nbmax, Nbjoueur	INT[1]	4
Map	CHAR[x*y]	(pour une taille (5,7), X étant une case jouée, O une case non jouée, 3 une case gagnée par le joueur 3) <b>OXO3O00O3O00O3O0OX3O0OX3OXO00000XOO</b>

Remarque : un retour à la ligne dans la colonne message des protocoles suivants sera en réalité un espace dans le buffer.

### Carte des machines physiques



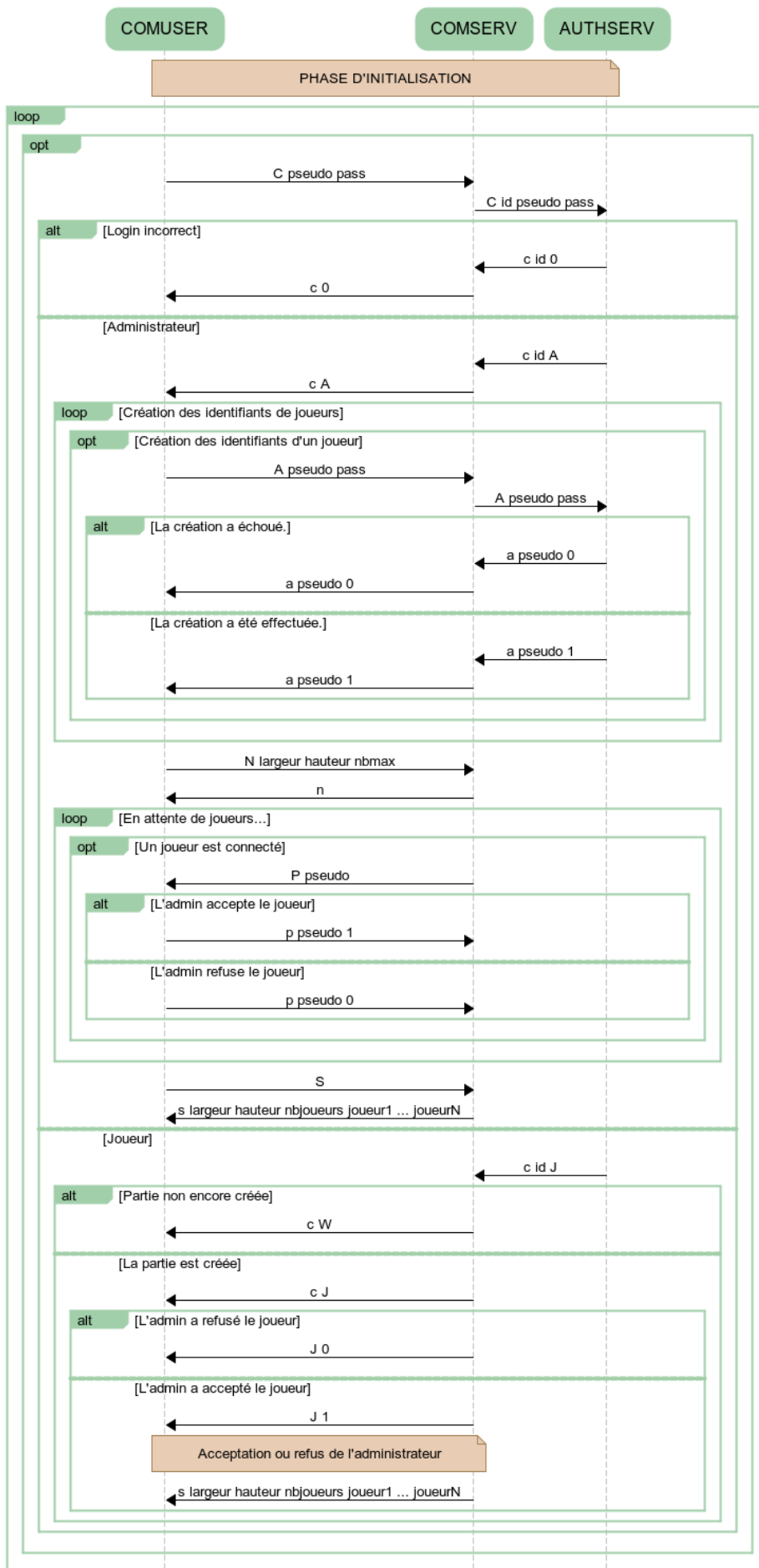
Pour des raisons de sécurité, il serait judicieux de faire communiquer le serveur applicatif avec le serveur d'authentification de manière crypté (SSL).

# Phase d'initialisation

## Protocoles

DE	A	MESSAGE	SIGNIFICATION
COMUSER	COMSERV	C <i>pseudo</i> <i>pass</i>	Un utilisateur souhaite se connecter
COMSERV	AUTHSERV	C <i>id</i> <i>pseudo</i> <i>pass</i>	Le serveur de communication demande au serveur d'authentification de vérifier les identifiants
AUTHSERV	COMSERV	c <i>id</i> 0   A   J	Le serveur d'authentification donne à comserv le statut de l'identification (l'identifié est un admin, un joueur ou s'est fourvoyé dans ses identifiants)
COMSERV	COMUSER	c 0 A J W	Le serveur d'authentification prévient d'un échec ou d'une réussite d'identification (admin A, joueur J). Il peut aussi répondre <u>attendre</u> (W) si aucun admin n'est connecté pour le moment.
ADMIN	COMSERV	A <i>pseudo</i> <i>pass</i>	L'administrateur crée les identifiants d'un nouveau joueur
COMSERV	AUTHSERV	A <i>pseudo</i> <i>pass</i>	L'administrateur crée les identifiants d'un nouveau joueur
AUTHSERV	COMSERV	a <i>pseudo</i> 0   1	Le serveur d'authentification confirme le succès (1) ou non (0) de la création d'un nouvel utilisateur.
COMSERV	ADMIN	a <i>pseudo</i> 0   1	Le serveur confirme le succès (1) ou non (0) de la création d'un nouvel utilisateur.
ADMIN	COMSERV	N <i>largeur</i> <i>hauteur</i> <i>nbmax</i>	L'administrateur crée une partie de taille <i>largeur</i> x <i>hauteur</i> pouvant contenir <i>nbmax</i> joueurs au maximum INT (2 caractères séparé par des espaces)
COMSERV	ADMIN	n	Le serveur confirme qu'il a reçu le message.
COMSERV	ADMIN	P <i>pseudo</i>	Le serveur prévient l'admin qu'un joueur vient de se connecter, afin de savoir s'il peut participer ;
ADMIN	COMSERV	p <i>pseudo</i> 0   1	L'admin choisit si le joueur peut participer (1) ou non (0).
COMSERV	COMUSER	J 0   1	Le serveur prévient l'utilisateur s'il a été accepté ou non dans la partie.
ADMIN	COMSERV	S	L'administrateur demande que la partie commence.
COMSERV	ADMIN + COMUSER	s <i>largeur</i> <i>Hauteur</i> <i>nbjoueurs</i> <i>joueur1</i> ... <i>joueursn</i>	Le serveur annonce le début de la partie.

## Diagramme de séquence



Ce diagramme de séquence est disponible en taille réelle en annexe. (dss/Initialisation.png)

## Phase de jeu

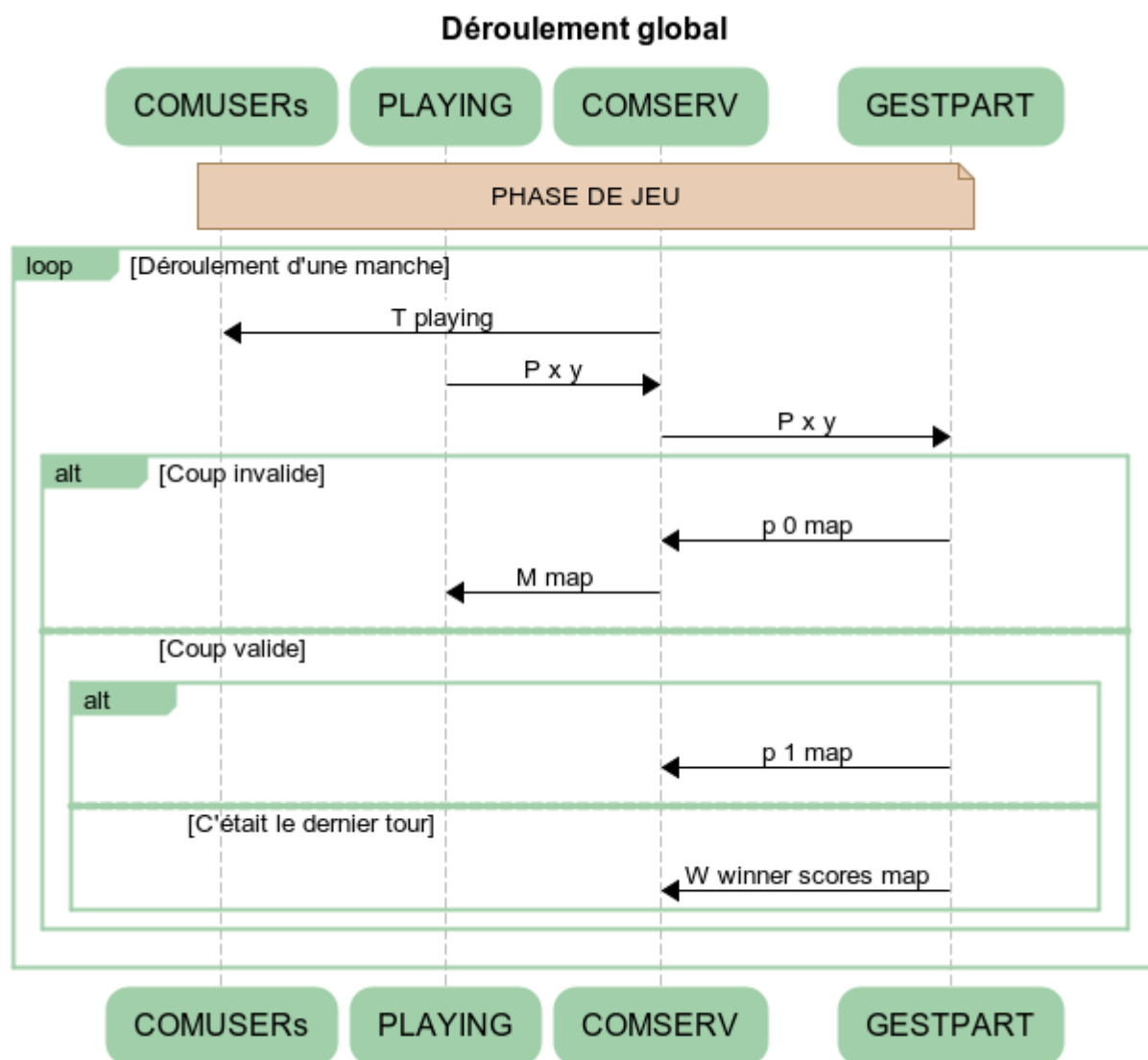
### Protocoles

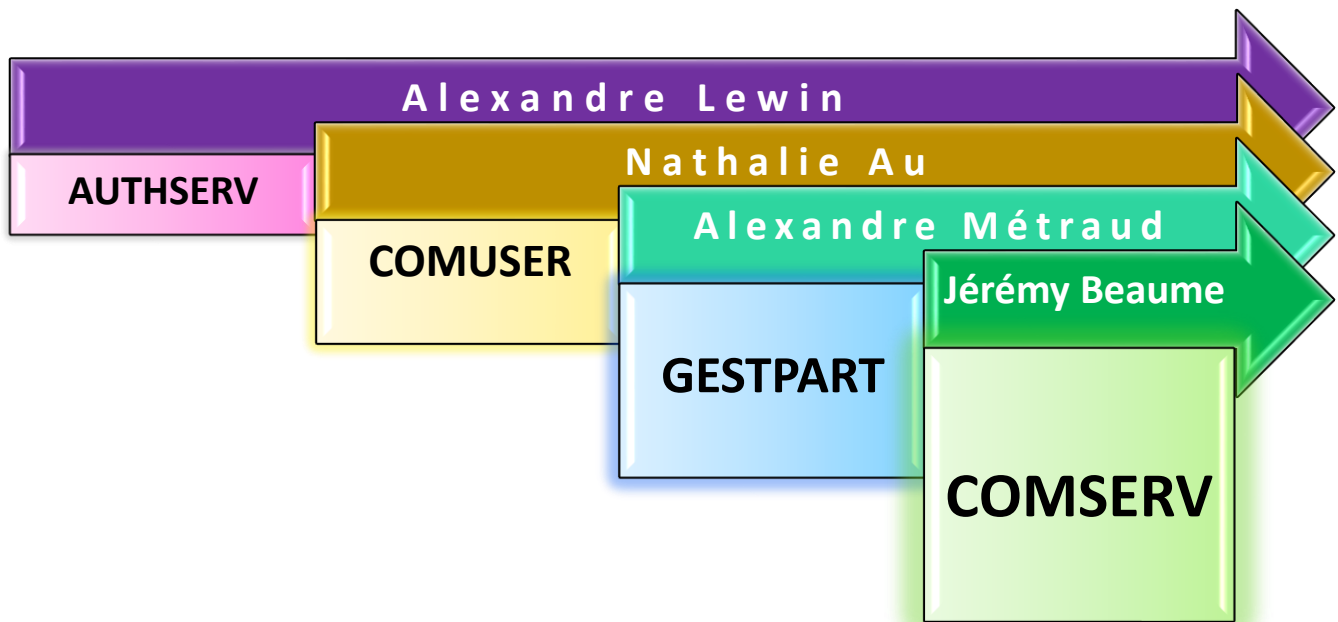
On nomme PLAYING le COMUSER qui doit jouer.

*map* est envoyé comme y mots de x caractères correspondant aux cases (et leur statut)

DE	A	MESSAGE	SIGNIFICATION
COMSERV	GESTPART (interne)	partie* init( largeur, hauteur, nbjoueurs)	Le serveur indique si le coup est valide et envoie la map.
COMSERV	COMUSERs	T playing score map	Le serveur prévient que c'est au joueur PLAYING de jouer et envoie la map et le score
PLAYING	COMSERV	P x y	Le joueur dont c'est le tour envoie les coordonnées de son coup.
COMSERV	GESTPART (interne)	int play (partie*, x, y)	Le serveur envoie le coup du joueur au gestionnaire de partie (Renvoie : -1 si le coup est incorrect, 0 si la partie est terminée, le numéro du joueur suivant si le coup est correct)
COMSERV	PLAYING	M map	Le serveur prévient que son coup est mauvais et envoie la map.
COMSERV	COMUSERs	W winner scores map	La partie est terminée, le gagnant est <i>winner</i>

**GESTPART** contiendra aussi destroy, (char\*) getmap (partie\*) et (int\*) getscore (partie\*)





Les différents serveurs et modules étant de complexités différentes, chaque intervenant travaillera sur un module différent et, une fois son module terminé, rejoindra le groupe de travail suivant. Alexandre Lewin réalisera AUTHSERV puis rejoindra Nathalie Au travaillant sur COMUSER ; une fois COMUSER réalisé, ils rejoindront ensuite Alexandre Métraud sur GESTPART ; une fois GESTPART réalisé, ils rejoindront Jérémy Beaume sur COMSERV.

On a ainsi des tâches bien réparties tout en ayant une efficacité de travail en groupe, sans aucune personne sous-utilisée durant une période.