

Intergiciels pour les Jeux en Réseaux

Patrice Torguet

torguet@irit.fr

Université Paul Sabatier

Plan du cours

- Introduction
- Intergiciels généralistes
 - RPC, RMI, CORBA, ICE
- Intergiciels spécifiques
 - Direct Play, XNA, TorqueNL, Raknet, Unreal
 - HLA, Openskies, Vega Prime, Delta 3D
 - Multiverse, 3DVIA Virtools, Unity 3D...
 - NeL, BigWorld, Hero Engine
- Conclusion

Introduction

- Middleware/Intergiciel : couche logicielle qui se place entre l'application et le système d'exploitation
- Comme le SE, il offre des services utilisables par plusieurs applications
- Comme une application, il n'est pas indispensable à toutes les applications et il s'appuie sur des fonctionnalités du SE

Introduction

- Il existe plusieurs types d'intergiciels (SGBD, communication, graphique...)
- Ici : intergiciels réseaux
- En général : ils utilisent les sockets (et éventuellement les threads et la synchronisation du SE) pour offrir des services de plus haut niveau

Introduction

- Exemples de services offerts
 - Appels de procédures/méthodes à distance
 - Nommage/localisation/découverte
 - Migration/Equilibrage des charges
 - Transferts de messages/événements/notifications

RPC : Remote Procedure Call

- Modèle client/serveur
- Appel de procédures à distance entre un client et un serveur
 - Le client appelle une procédure locale (souche – stub – ou proxy)
 - La procédure locale utilise une connexion socket pour envoyer un identifiant de procédure et les paramètres au serveur
 - Le serveur reçoit la requête grâce à un socket, extrait l'id de procédure et les paramètres (ceci est fait par un squelette – skeleton)
 - Le squelette exécute la procédure et renvoie le résultat via le socket
- Outil rpcgen
 - Génère la souche et le squelette à partir d'un fichier présentant l'interface des méthodes dans un format indépendant du langage (RPCL : RPC Language)

RPC : Remote Procedure Call

- Couche de présentation XDR (eXternal Data Representation)
 - Format pivot de représentation des données de types primitifs et structurés (tableaux, tableaux de taille variable, structures...)
 - Indépendant de
 - L'architecture (little endian/big endian)
 - Le langage (ordre ligne/colonne dans les tableaux C et Fortran)
 - Le système (ASCII, EBCDIC)
- Limitation :
 - Pas de gestion des concepts objets (encapsulation, héritage, polymorphisme)
 - Pas de services évolués : nommage ...
- Successeurs :
 - RMI : mono langage, multi plateforme
 - CORBA : multi langage, multi plateforme
 - COM : multi langage, mono plateforme (multi pour DCOM)
 - SOAP / .NET / web services : multi langage, multi plateforme

RMI : Remote Method Invocation

- RPC en Java
 - Invocation de méthodes sur des objets distribués
 - Très lié au langage => simple à mettre en œuvre
- Outils
 - Génération des souches/squelettes
 - Service de nommage simple (précise la machine d'exécution)
 - Activation
- Mono langage et multi plateforme : de JVM à JVM
- Orienté objet : utilisation de la sérialisation
- Dynamique : téléchargement des classes des souches et des paramètres via HTTP (<http://>) ou NFS/SMB (<file://>)
- Sécurisé : SecurityManager et fichier de sécurité .policy

CORBA : Common Object Request Broker Architecture

- RPC orientés objets + nombreux services
 - Invocation de méthodes sur des objets distribués
 - Indépendant du langage => utilisation d'un IDL (Interface Definition Language)
- Outils
 - Génération des souches/squelettes depuis l'IDL
 - Service de nommage complet, service de trading (annuaire page jaunes)
 - Gestion des événements, de la persistance, du temps-réel, du streaming...
- Multi-langage et multi-plateforme
- Interopérabilité entre plusieurs ORBs : GIOP/IIOP

ICE - Internet Communications Engine

- Middleware similaire à CORBA (basé sur un IDL, multiplateforme et multi-langage) mais beaucoup plus simple à déployer, à utiliser et très efficace
- Multi-langage : C++, Java (y compris sur Android), .NET, Python, PHP, Ruby, et Objective-C (y compris sur iPhone)
- Multiplateforme: Win, Linux, OS X, Solaris, Android, iOS
- Gère des communications TCP, UDP, UDP multicast, SSL en IPv4 et IPv6

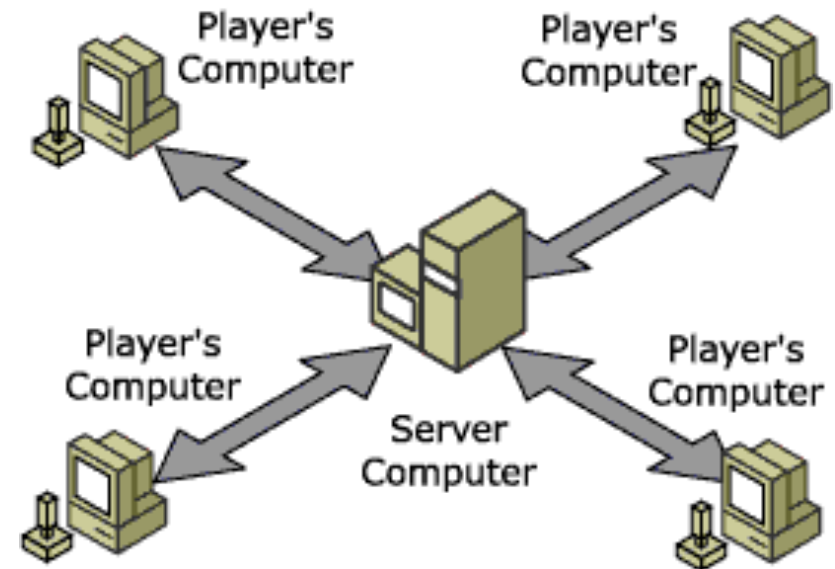
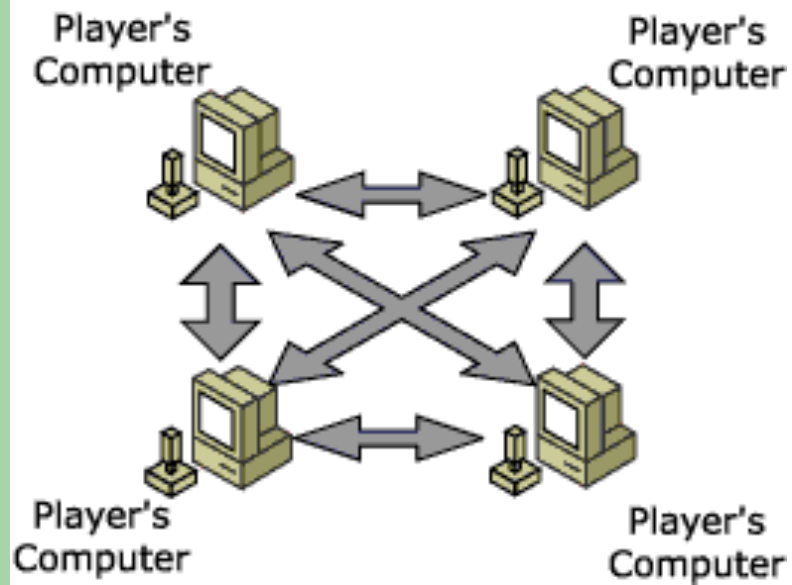
ICE - Internet Communications Engine

- Services : RPC synchrones, asynchrones, unidirectionnelles (oneway), à traitement par lot, invocation dynamique, retransmission de RPC, traversée des firewall et NATs, persistance des objets dans une base de donnée, évènements, auto-patcher, compression, gestion des versions...
- A été utilisé pour faire un MMO (Wish de Mutable Realms – annulé avant la sortie)
- Disponible sous GPL à : <http://www.zeroc.com/>

Direct Play

- Fait partie de Direct X (v3 – v9 - abandon : v10 => XBox Live, Games for Windows Live et XNA)
- Support de 2 architectures
 - Egal à égal totalement connecté
 - Client/Serveur
- Filtrage par groupes géré par le serveur

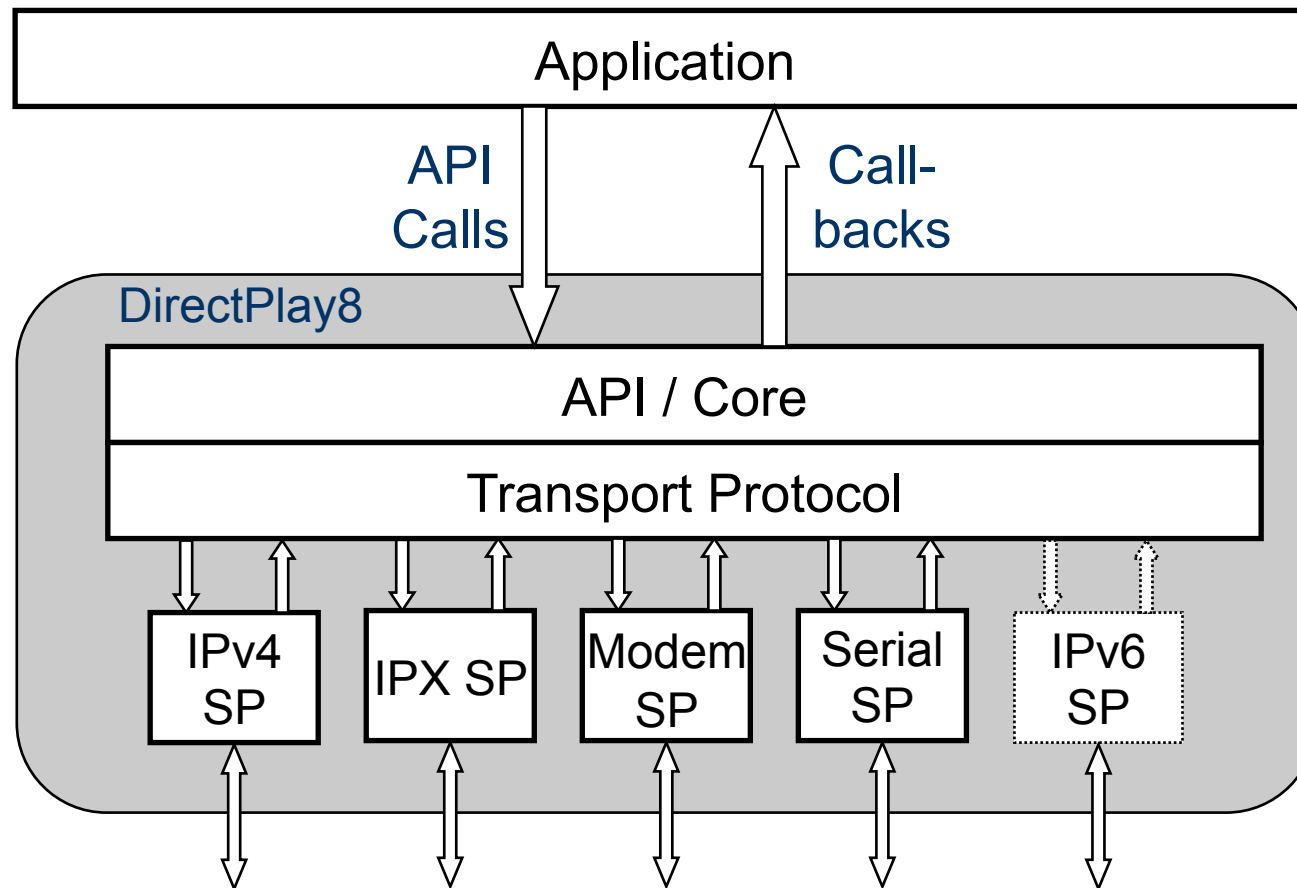
Direct Play : architectures



Direct Play

- Audio-conférence multi-utilisateur
- Outils de test
 - Netmon (monitoring temps-réel)
 - Dp8Sim (simulation de bande passante limitée et de pertes)

Direct Play : architecture en couches



XNA Game Studio

- Multiplateforme : (Win), Xbox, (Windows Phone)
- Langage : C#
- Lobby avec Games 4 Windows/Xbox LIVE
 - Création de parties, recherche de parties, mise en relation des applications, profils utilisateur, scores...
- Mode de jeu : local (écran partagé), en réseau local ou sur Internet via Xbox LIVE
- Architectures répartie : P2P et C/S avec un hôte responsable de la connexion LIVE

XNA Game Studio

- Gestion QOS : best effort, ordonné, fiable, les 2, données de chat (indépendant du reste et non crypté)
- Gestion de la voix
- Simulation de bande passante, perte et latence incluse dans XNA
- Gratuit pour les jeux sous Windows (mais pas réseau), payant (\$99) pour développer sur Xbox LIVE (Indie) ou Windows Phone (mais pas réseau)
- Infos et téléchargement : <http://create.msdn.com/>

TorqueNL

- Issu des jeux “Starsiege: TRIBES” et TRIBES 2
- Architecture Client/Serveur orientée FPS
 - Compression des données à envoyer
 - Au niveau du bit (booléens, entiers en précisant le nombre de chiffres, flottants en précisant la précision)
 - Plusieurs QoS
 - Ordre et fiabilité
 - Fiabilité seule
 - Non fiable
 - Dernière version uniquement (lors des ré-émission on envoie la version à jour)
 - Aussi vite que possible (données ré-émises jusqu’à acquittement)

TorqueNL

- Interpolation/Extrapolation et Dead-Reckoning
- Filtrage des données non intéressantes pour un client (l'application peut préciser si un objet est visible ou non visible d'un client)
- Gestion de priorités pour les mise à jour d'objets
- Emission des données statiques à la connexion
- RPC
- Cryptage symétrique et asymétrique
- Gestion des firewall et des NAT
- Gestion d'un serveur maître pour rechercher des parties
- C++ sous GPL ou en version commerciale (?)
- Disponible à <http://opentnl.sourceforge.net/> et <https://github.com/nardo/tnl2> (v2)

Raknet

- Client/Serveur ou Peer to Peer
- Multiplateforme : Windows, Linux, OS X, iOS, Xbox et PS3
- Fonctionnalités :
 - Compression des messages
 - Plusieurs QoS (fiable, séquence, ordre)
 - Cryptage
 - Gestion des mises à jour (auto patcher)
 - Gestion de la voix sur IP (compression ~ 2 kbps)

Raknet

- Système de réplication d'objets (sérialisation)
- RPC
- Evènements en réseau (signaux et slots)
- Emulation de réseaux (latence, bande passante limitée)
- Serveur maître/lobby spécifique ou utilisation de Steam, Xbox LIVE ou PlayStation Network
- Gestion des firewall/NAT
- C++/C# - License Indie gratuite + licence commerciale
- <http://www.jenkinssoftware.com/>

Unreal

- Moteur de jeu de Unreal
- Client/serveur FPS
 - Réplication d'objets avec priorités
 - QoS multiples
 - Filtrage des objets non visibles (lien avec arbre BSP)
 - RPC
 - Quantification pour compresser les données
 - Interpolation/Extrapolation
- Utilisable avec UnrealScript (gratuit pour les mods)
- Licence commerciale
- Doc : <http://unreal.epicgames.com/Network.htm>

HLA (High Level Architecture)

- Motivations

- Permettre l'interopérabilité et la réutilisabilité des simulations existantes et futures
- Définir une architecture fonctionnelle qui puisse s'implémenter sur plusieurs architecture matérielles, logicielles et réseaux
- Tenter de résoudre les problèmes des standards précédents comme DIS

- Disponibilité

- Documentation disponible à : <http://hla.dmsomil/>
- Version opensource du logiciel : <http://savannah.nongnu.org/projects/certi/>

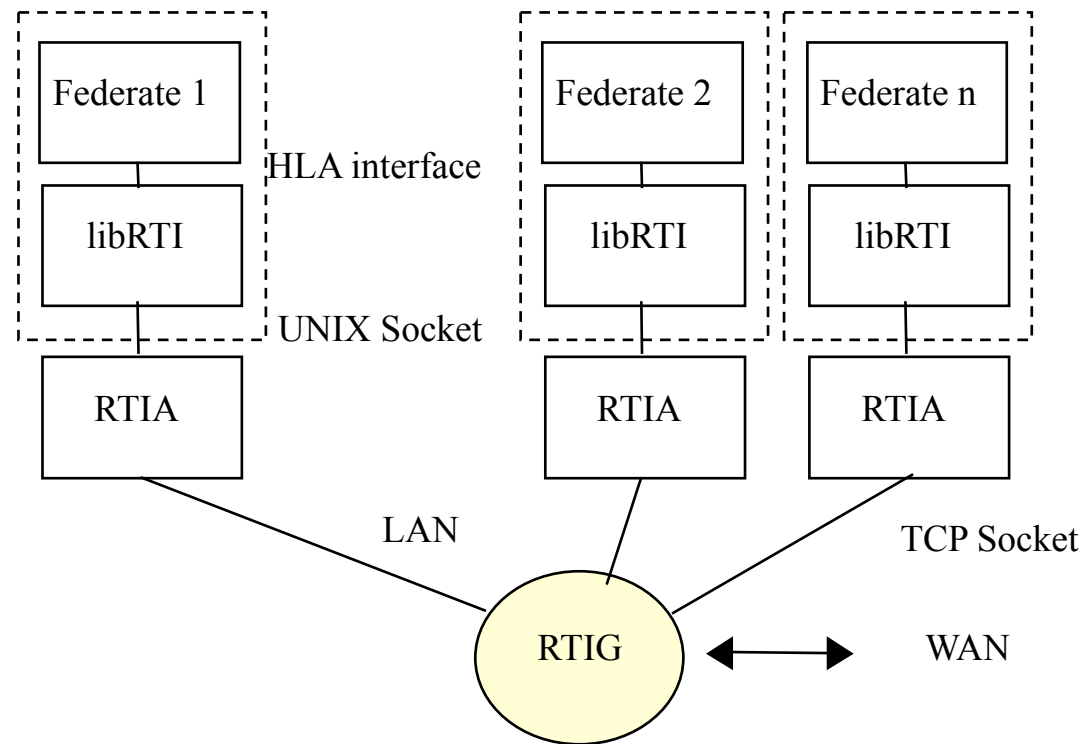
HLA (High Level Architecture)

- Composants

- Règles de fédération : principes généraux qui doivent être appliqués lors du développement d'une fédération compatible avec HLA
- RTI (Run-Time Infrastructure) : une infrastructure d'exécution distribuée qui gère toutes les communications dans une fédération
- OMT (Object Modeling Template) : modèle orienté-objet générique qui permet de définir les composants d'une fédération

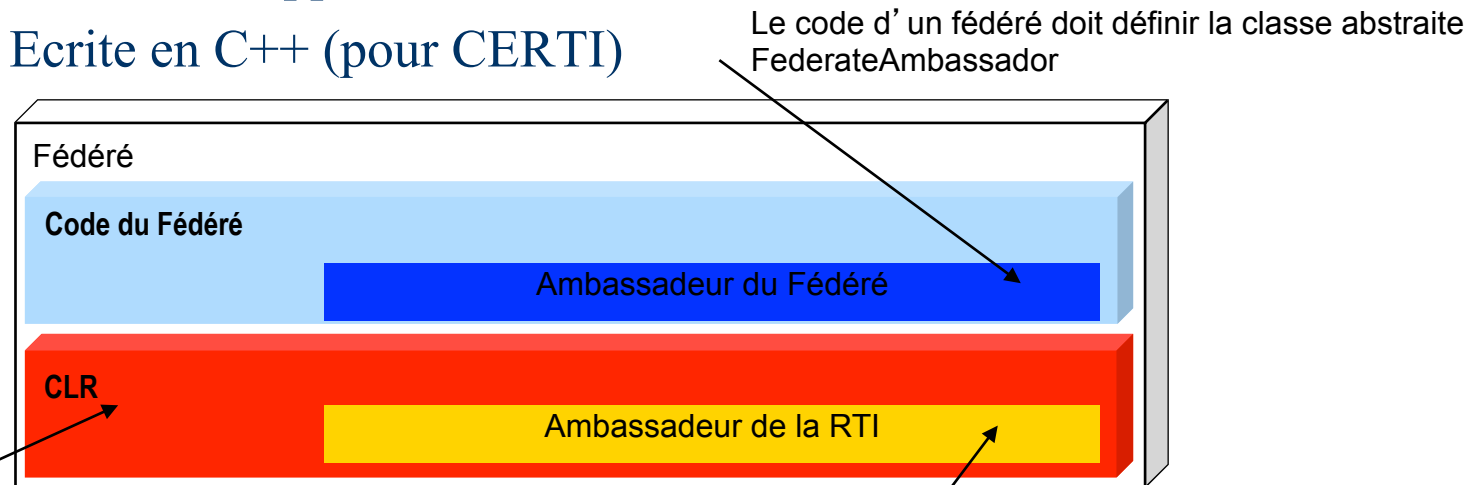
HLA (High Level Architecture)

- Exemple d'architecture (CERTI)

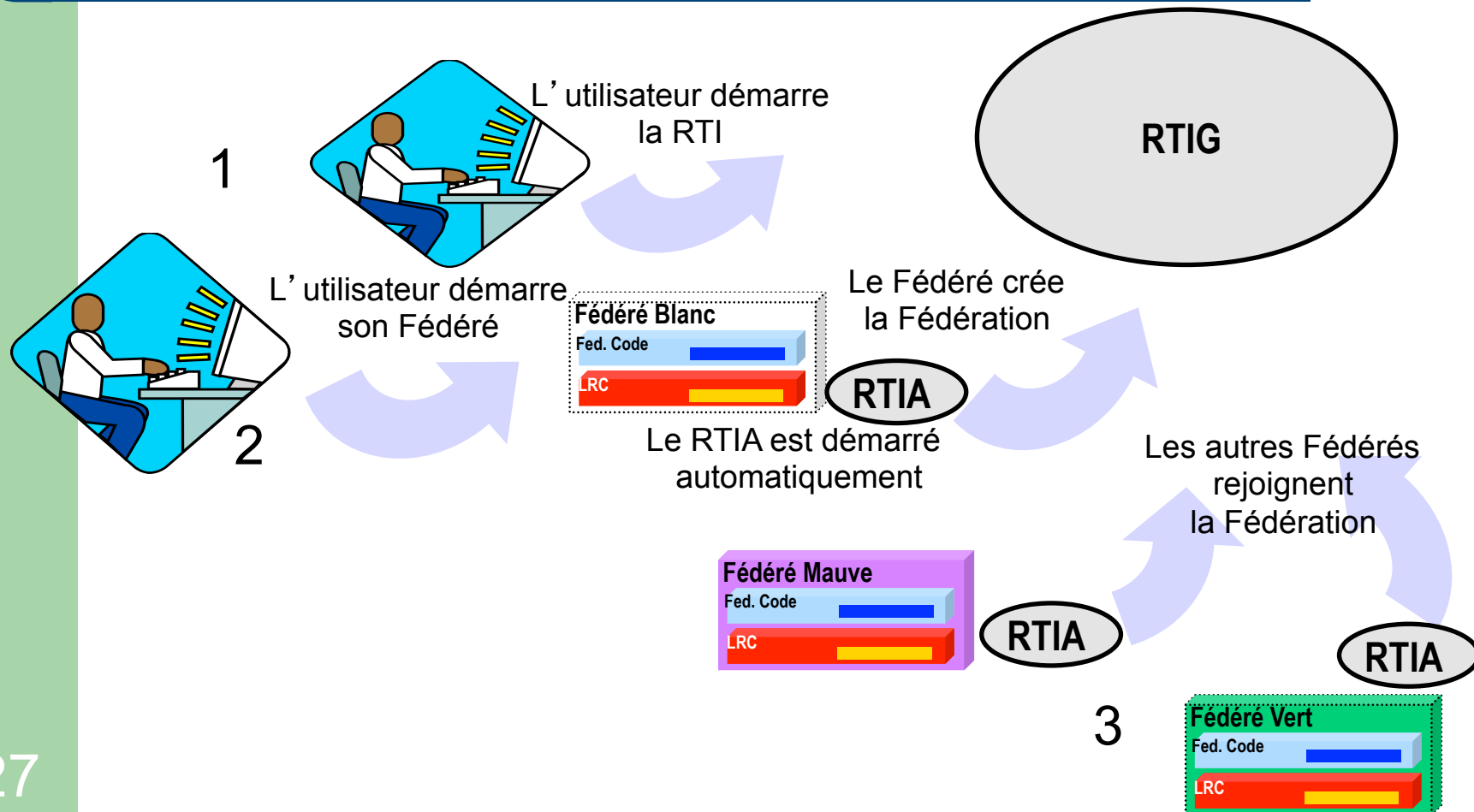


HLA (High Level Architecture)

- La bibliothèque de classes (libRTI)
 - Offre des services aux fédérés
 - Le fédéré communique avec les RTIA, RTIG et les autres fédérés au travers de l'appel de méthodes de libRTI
 - Ecrite en C++ (pour CERTI)



HLA (High Level Architecture)



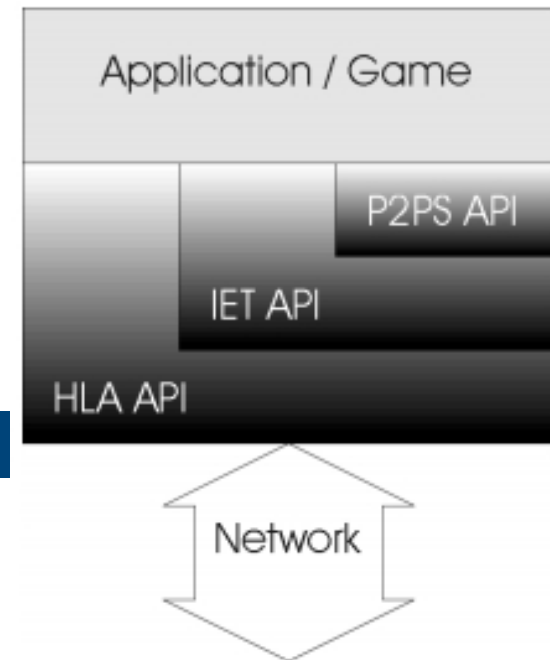
HLA (High Level Architecture)

- Exemple de base du code d'un fédéré
 - 1. Créer une fédération si elle n'existe pas
 - 2. Rejoindre la fédération
 - 3. Déclarer les données de la fédération qui doivent être publiées et celles auxquelles on veut s'abonner
 - 4. Envoyer et recevoir des données vers/depuis les autres fédérés
 - 5. Quitter la fédération
 - 6. Détruire la fédération si elle est vide
- Autres RTI (commerciales)
 - pRTI de pitch.se
 - Mäk RTI de Mäk (mak.com)

Openskies

- Plusieurs API

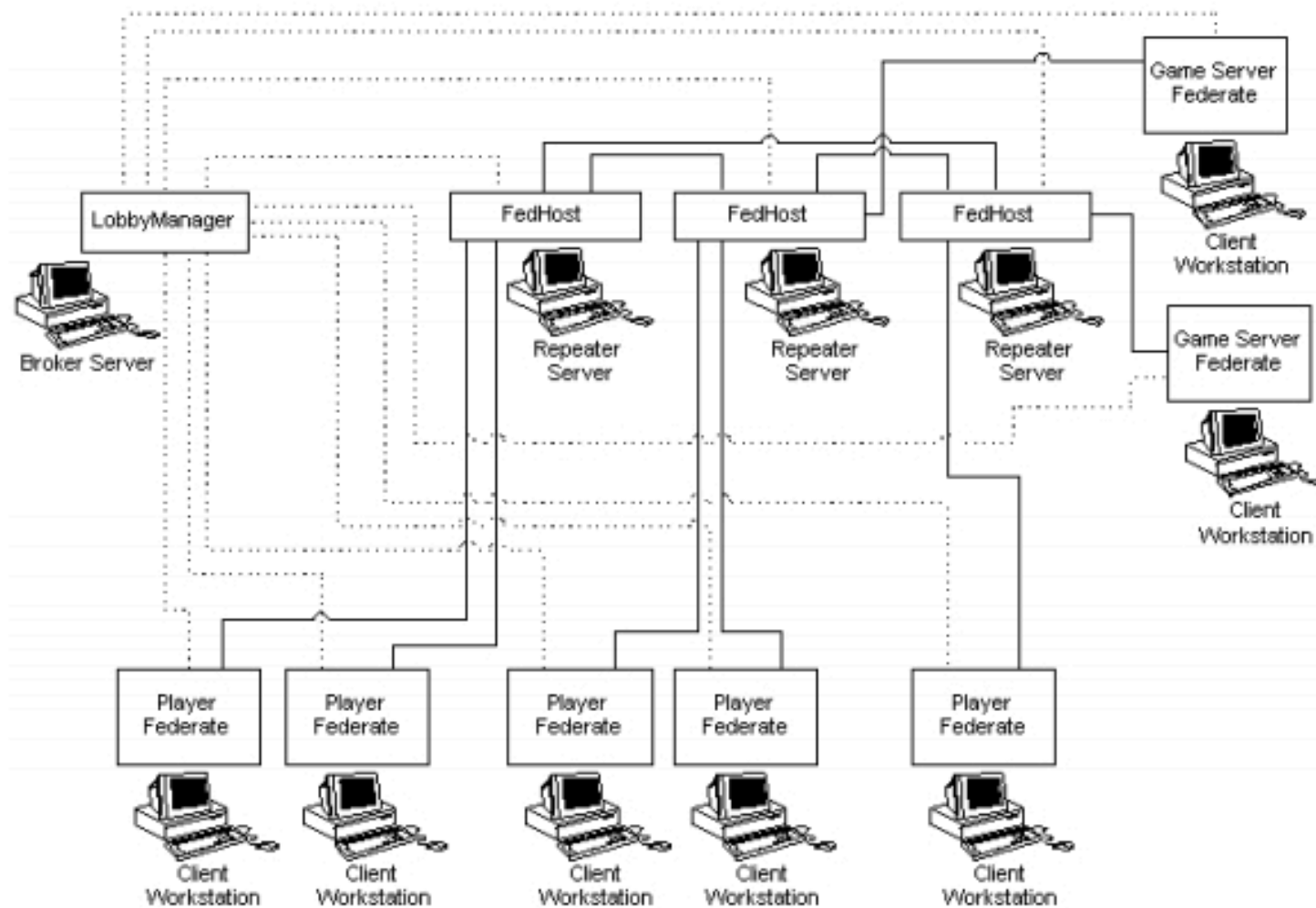
- bas niveau : HLA (RTI)
- IET (Import Export Table) génère le code HLA pour mettre à jour des objets
- P2PS (Point to Point Switch Messaging Interface) permet d'envoyer des messages d'un objet à un autre objet quelque soit leur localisation et même s'ils ne sont pas encore créés (bufferisation)



Openskies

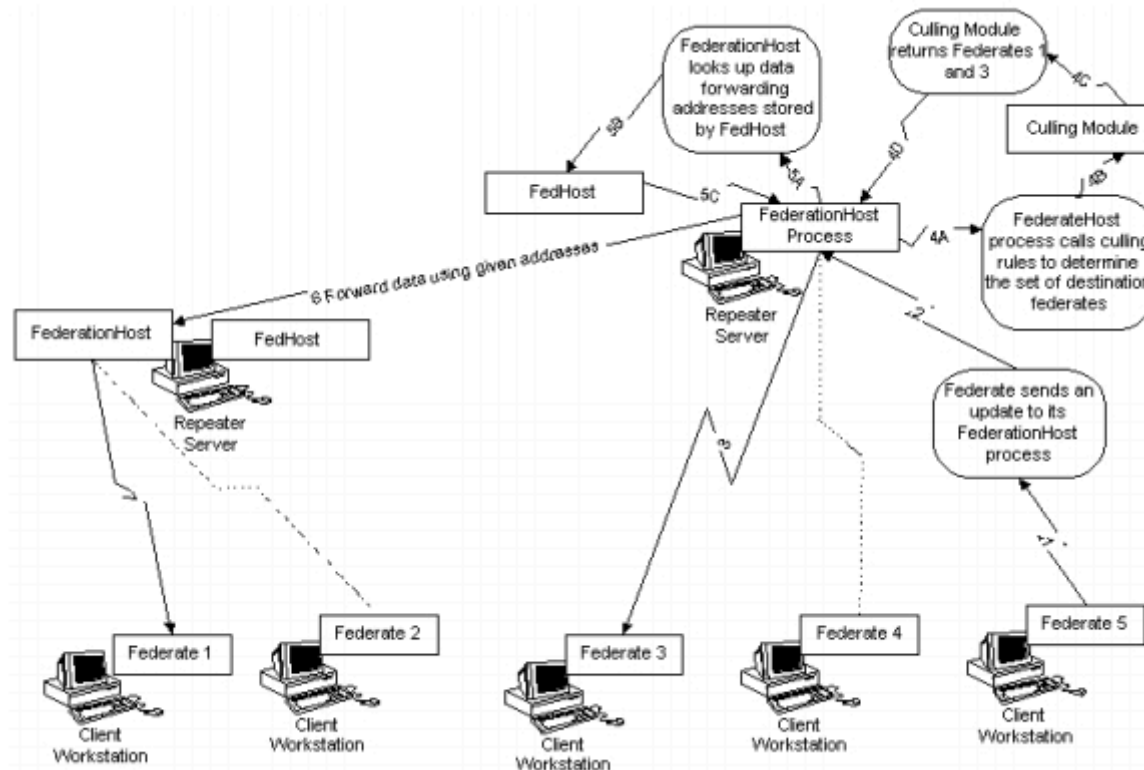
- Architectures
 - Architecture Peer to Peer ou client/multi-serveur
 - Serveurs linux dédiés (distribution linux spéciale)
 - 3 types de serveurs
 - LobbyManager (point d'entrée - authentifie - permet de trouver un FedHost)
 - FedHost (s'occupe du routage des messages entre fédérés)
 - Chacun gère jusqu'à 500 fédérés
 - Game Server (s'occupe du jeu : IA...)
 - Un ou plusieurs
 - Codés comme des fédérés

Openskies



Openskies

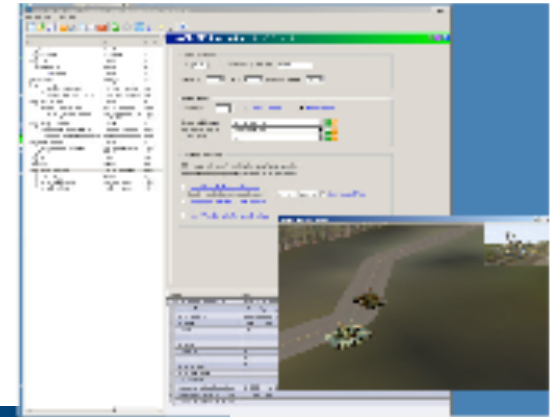
- API FedHost pour développer des modules de filtrage (culling)
 - Utilisation possible d'une grille 2D pour filtrer sur la position



Openskies

- C++ - Windows pour les fédérés
- Licence de développement gratuite, licence commerciale.
- <http://www.openskies.net/>

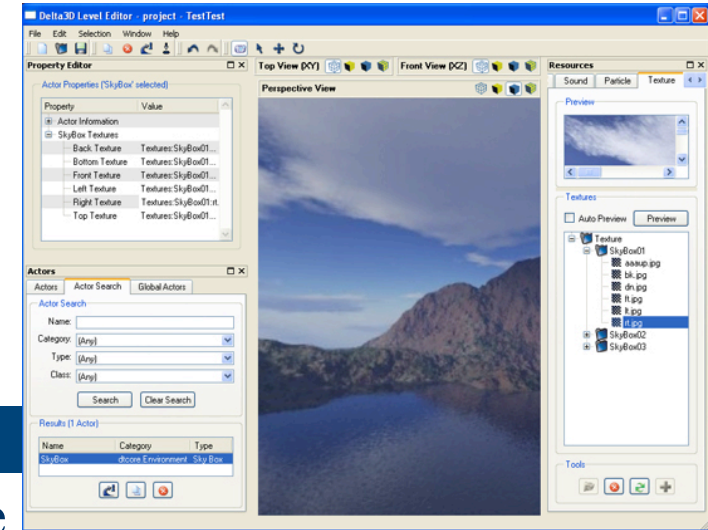
Vega Prime (HLA/DIS mod)



- Vega Prime
 - Application graphique pour générer des simulateurs 3D temps-réel sans codage
 - Nombreux modules (marine, effets atmosphériques, infra-rouge, radar...)
- HLA/DIS module
 - Permet de rajouter la distribution avec HLA en utilisant le RPR FOM (équivalent de DIS pour HLA) ou avec DIS
 - Sans codage
 - Basé sur VR-Link de Mâk (plate-forme de répartition fonctionnant avec DIS, HLA ou TENA)
- http://www.presagis.com/products/multigen_paradigm/details/vegaprime/

Delta 3D

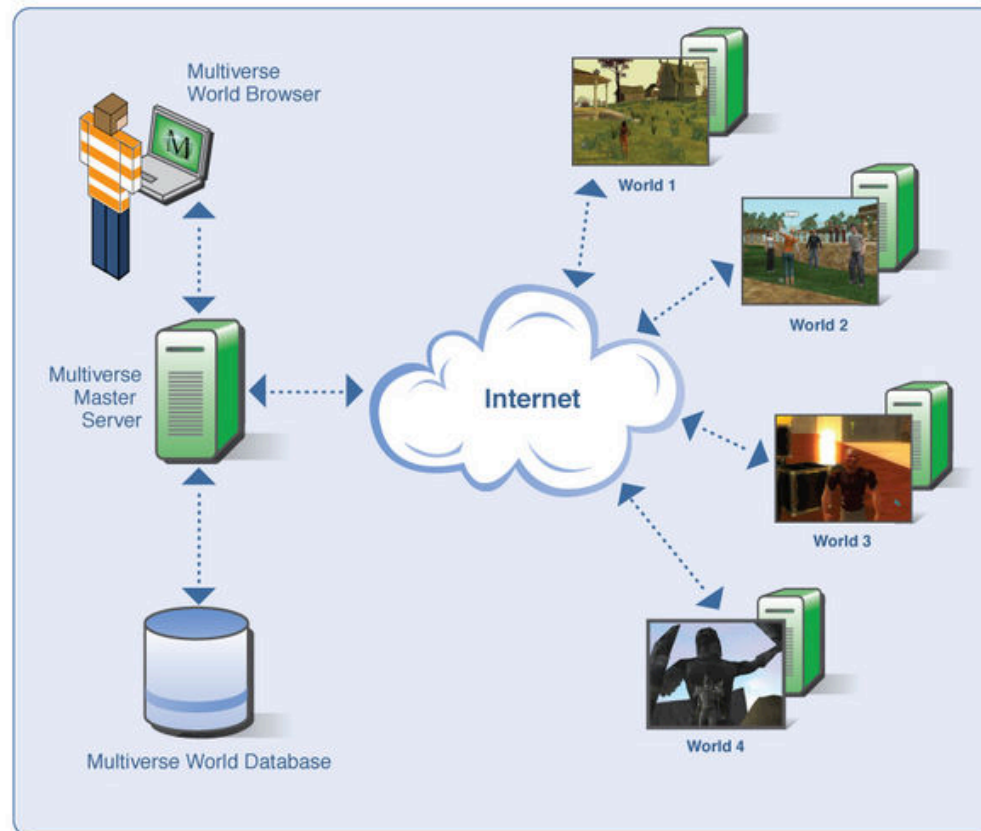
- “Clone” opensource de Vega Prime
- Basé sur de nombreux toolkits
 - Open Scene Graph (3D)
 - Open Dynamics Engine (ODE - moteur physique)
 - Open AL (audio)
- Classes de haut niveau
 - Toolkit Réseau spécifique
 - Encapsulation de HLA (RTI spécifique mais adaptable)
- Outil graphique pour gérer des simulations 3D temps-réel



Multiverse

- Environnement Client/Serveur(s) complet pour faire des MMORPG (?)
 - Gestion de quêtes, de compétences, combat, communication, inventaire et échanges, artisanat, économie, IA, facturation...
 - Outils pour concevoir le monde virtuel
 - World Browser (pour trouver les mondes)
- APIs pour customiser le client et le serveur
- Licence commerciale (10% des revenus du jeu)
- <http://multiverse.net/>

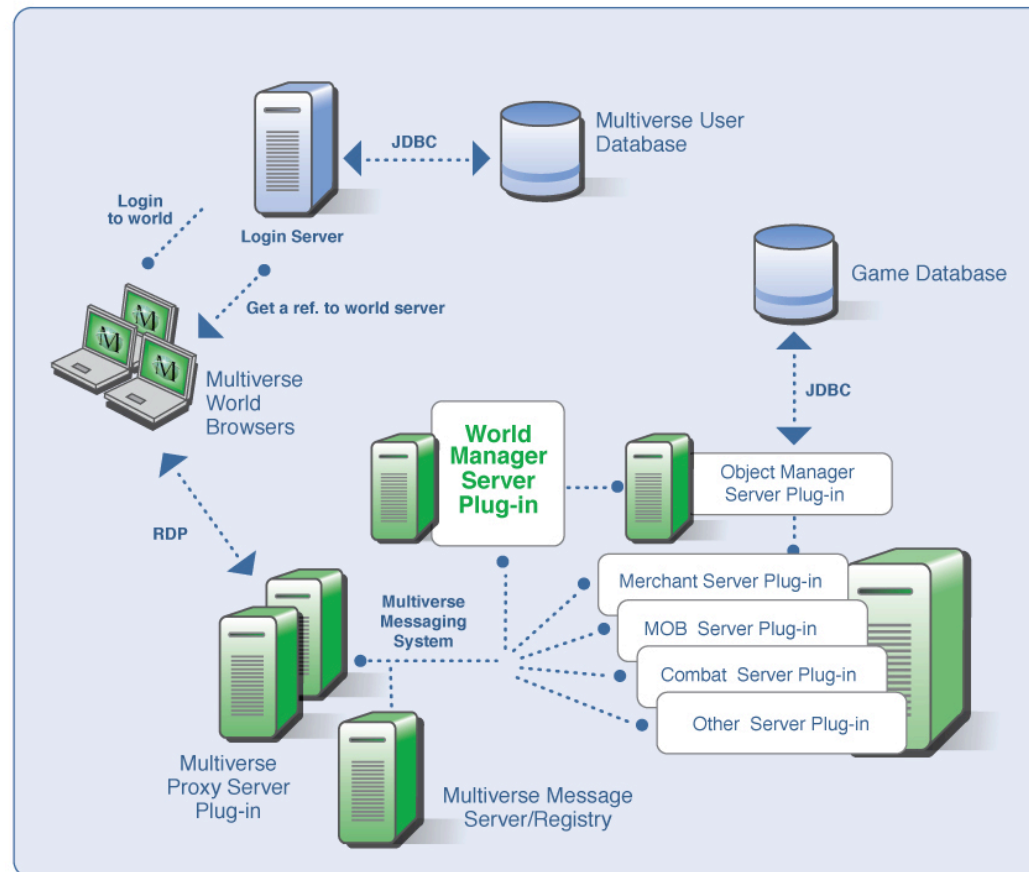
Multiverse



Multiverse

- Client écrit en C++ avec Ogre3D et customisable en Python (y compris l'interface graphique)
- Serveur écrit en Java
- JDBC pour la persistance
- architecture à plug-ins
 - plug-ins standards (tous optionnels)
 - API pour développer des plug-ins spécifiques

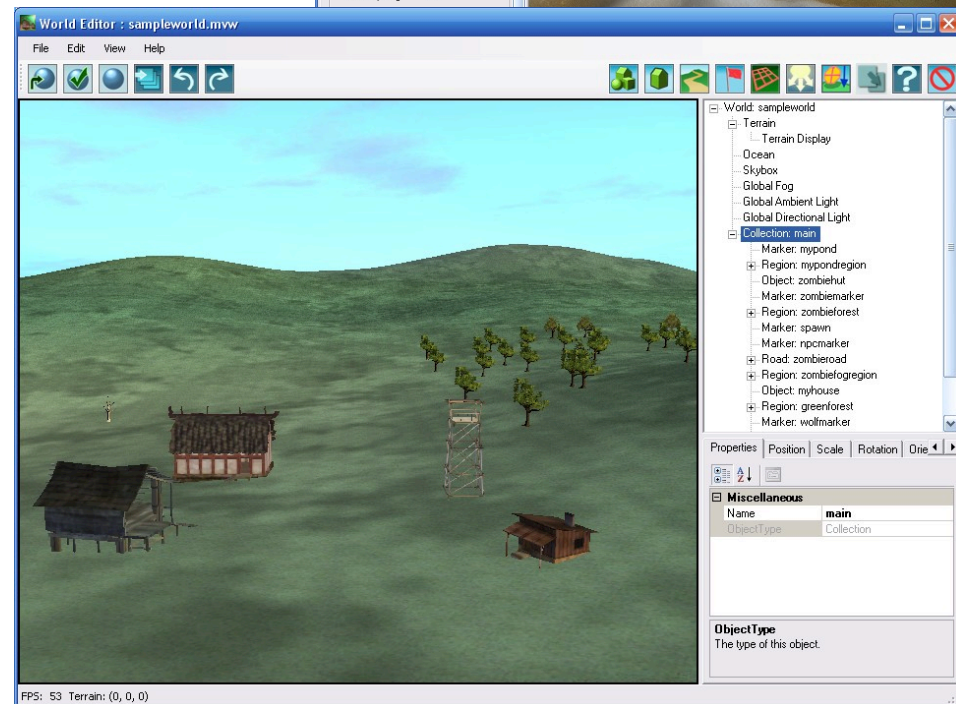
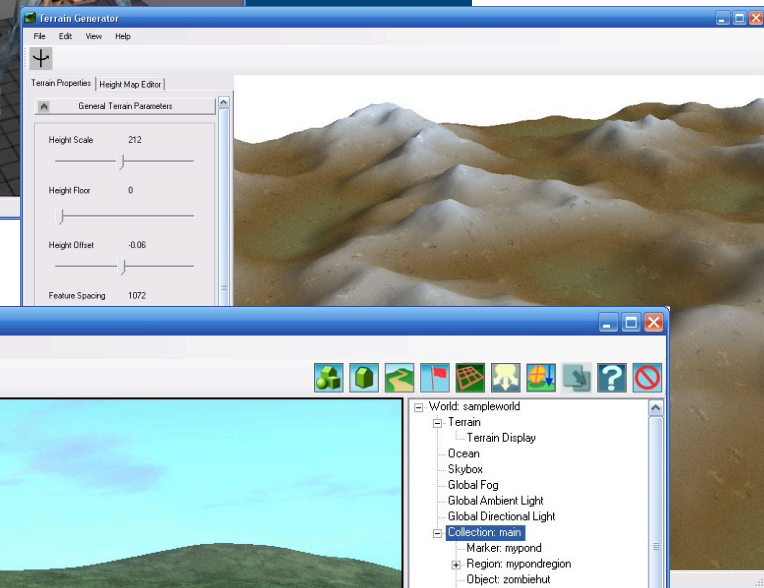
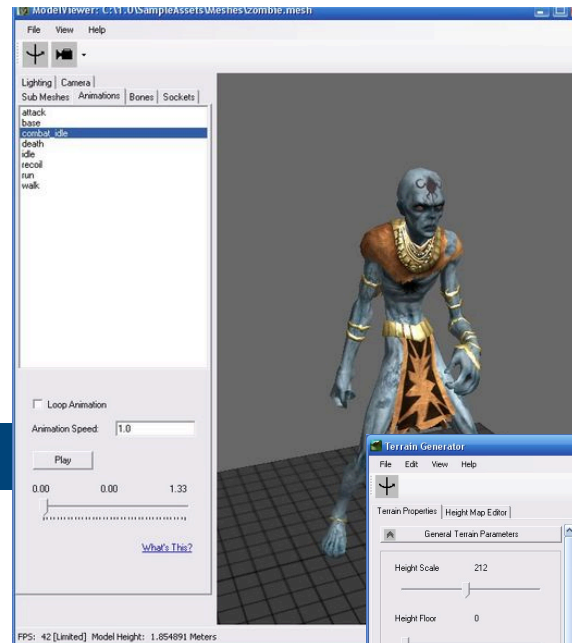
Multiverse



Multiverse

- Outils

- ModelViewer
- Terrain Generator
- WorldBuilder



3DVIA Virtools

- Virtools est un générateur d'applications 3D temps-réel
- <http://www.3ds.com/products/3dvia/3dvia-virtools/>

Virtools : D:\AB\Sales\avant_vente\par produits\WRNR\WPack\CATIA_session\Auto_rendering\3D Entities\compo.cmo

File Resources Editors Options NXN alienbrain Help

3D Layout

New Camera

Building Blocks Virtools Resources Variables Manager Hierarchy Manager

Category	Behavior Name	Apply to	T
Grids	Bezier Progression	Behavioral Object	Interpolates a
Interface	Chrono	Behavioral Object	Chronometer
Controls	Collection Iterator	Behavioral Object	Retrieves eac
Fonts	Counter	Behavioral Object	Exits 'Max Co
Layouts	Delayer	Behavioral Object	Waits for a gi
Primitives	Hierarchy Parser	3D Entity	Iterate recurs
Screen	IBCQ	Behavioral Object	Incremental E
Text	Linear Progression	Behavioral Object	Increments fr
Lights	Noise Progression	Behavioral Object	Interpolates a
Logics	Timer	Behavioral Object	Waits (throw
Array	While	Behavioral Object	Loops while V
Attribute			
Calculator			
Groups			
Interpolator			
Loops			
Message			
Streaming			
Strings			
Synchro			
Test			
Materials-Textures			
Mesh Modifications			
Multiusers			

Schematic Level Manager Shader Editor

Show All Hide All Hide Selected Hide Other Trace Delete All Marks

ReplaceMaterial

Test Door open

Body_part.1_SPO0

Key NA

Key Waiter

Bezier Progression

Interpolator Object Orientation

In

4

In No action X: NA Y: NA Z: NA Script added to: Body_part.1_SPO0 NA FPS

3DVIA Virtools

- Fonctionnalités
 - Importation d'objets
 - Gestion des animations
 - Création de scripts graphiques (building blocks et flux)
 - SDK pour développer des bb ou une application complète
 - plugin pour le web
- Multiuser Server
 - Permet de gérer les aspects répartis
 - Architecture : peer server (un client gère les aspects serveur) ou client/serveur
 - Gère du dead-reckoning
 - Peu de détails sur l'implémentation

Unity 3D et équivalents

- De nombreux équivalents à Virtools existent. La plupart utilisent un langage de script à la place du visual programming :
 - Unity 3D : utilise C# ou javascript, pour Mac, PC, iPhone, Android, Wii, gratuit pour des jeux indépendants, licence commerciale (\$~1500) pour certaines options (ombres temps-réel, fonctionnement sur téléphone...)
 - Shiva 3D : utilise lua, environnement de dev pour PC, exec sur PC, Mac, Linux, iPhone, Wii, version gratuite pour s'entraîner, licence commerciale (€~3000)
 - Quest3D, Nova (Vertice), WorldWiz...

Ryzom Core/NeL

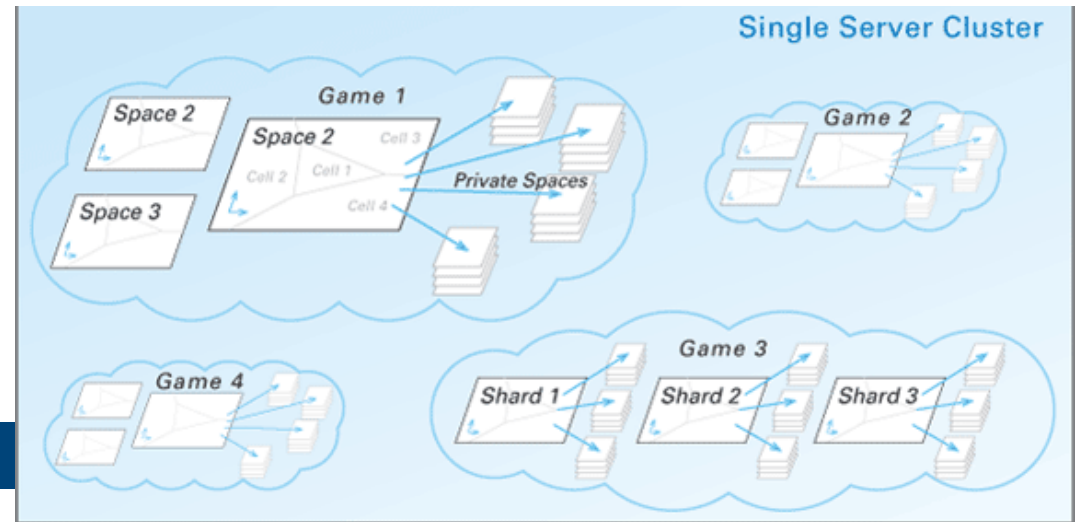


- Moteur pour MMO
- Développé pour le jeu Ryzom (<http://www.ryzom.com>)
- Toutes les bibliothèques sont en AGPL v3
 - 3D, réseau, IA, moteur physique simple...
- Toutes les ressources 3D et sons sont en Creative Commons
- NelNet
 - Propose une architecture à “shards” typique
 - Chaque shard a un serveur d’administration et 1 ou plusieurs serveurs de jeu qui font fonctionner un ensemble de services (persistance, IA, sessions, proxy...)

Ryzom Core/NeL

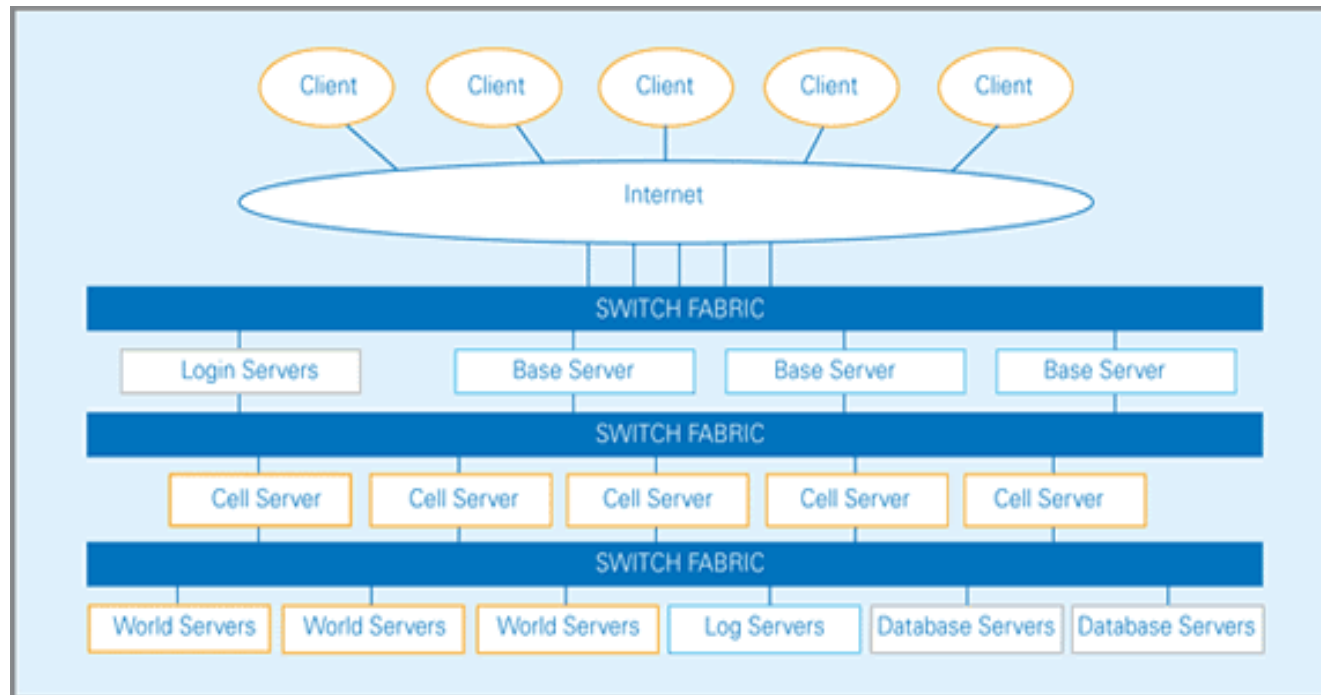
- C++ pour Linux/Windows (serveur) et Linux/windows/OS X (client)
- http://dev.ryzom.com/wiki/ryzom/NeL_Net

BigWorld



- Moteur de MMO clé en main
- Livré avec un MMO complet à customiser
- Adopté par un grand nombre de MMO en développement comme World of Tanks
- Côté serveur
 - équilibrage des charges dynamique
 - re-découpe les zones selon la charge
 - gère de la tolérance aux pannes
 - permet de gérer des instances et des shards voire plusieurs jeux sur un même cluster

BigWorld

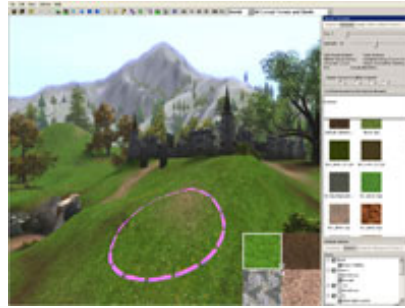


BigWorld



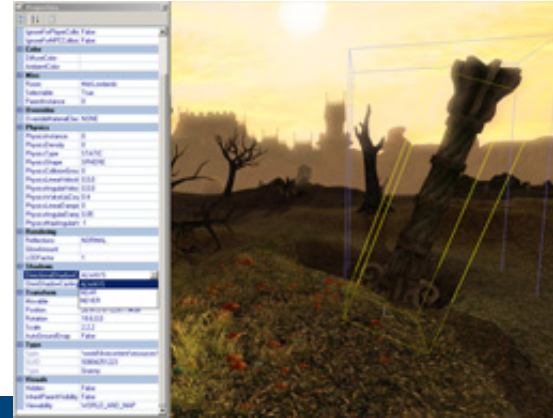
- Permet d'ajouter et retirer dynamiquement des serveurs au cluster
- Gère des priorités et des niveaux de détails (quantification ?) pour les données à transférer de façon à gérer au mieux la bande passante disponible
- Codé en C++ mais contrôlable via Python
- Offre des outils de monitoring
- Côté client
 - Gestion de Windows et Xbox 360 (?)
 - Moteur 3D spécifique de bon niveau
- Outils pour produire du contenu
- <http://www.bigworldtech.com/>

BigWorld



- Outils pour produire du contenu
 - Editeur de mondes : terrain, flore, construction de bâtiments simplifiée. Outil collaboratif.
 - Editeur de modèle : voir des modèles exportés avec le moteur de jeu, éditer les animations
 - Editeur d'effets de particules
- Licence commerciale (plusieurs centaines de milliers de dollars), Indie (\$299/an sans C++, \$2999 avec C++), education (\$299/an sans C++)
- <http://www.bigworldtech.com/>

Hero Engine



- Solution similaire à BigWorld
- Outils similaires mais plus interactifs/collaboratifs
- On édite le MMORPG en utilisant le client du MMORPG
- Langage de script HSL
- Utilisé pour Star Wars: The Old Republic de BioWare
- On peut se connecter gratuitement à une Sandbox pour tester le moteur
- <http://www.heroengine.com/>

Hero Engine

- Les machines serveur font tourner un ensemble de processus qui sont soit :
 - Uniques : compilateurs de scripts, gestion des stats, coordinateur, annuaire des services, gestion de login...
 - Multiples : serveurs de zones, proxies, simu physique (PhysX), contenu...
- Voir schéma dans le wiki: <http://hewiki.heroengine.com/>
- Minimum : 1 serveur CentOS avec Oracle (Quad Core Xeon 2GHz, 8 Go RAM, 4 To) + 1 serveur Win Server 2k8 pour le reste (Quad Core Xeon 2 GHz, 8 Go RAM, 1 To)

Hero Engine

- HeroCloud
 - La société gère les serveurs
 - ~ \$5000 par équipe (max 25 dev)
 - ~\$2000/an (1ère année offerte)
 - 15% de Royalties
- Achat complet possible (probablement au même prix que BigWorlds)



Jeux et chatworld customisables

- Jeux customisables
 - Quake (réalité augmentée : ARQuake, IA : bots - idem Unreal)
 - ORTS (moteur de RTS pour tester des algos d'IA)
 - Neverwinter Nights (enseignement de l'Anglais et des maths en Angleterre, du journalisme aux USA et pour des recherches en IA)
- Chatworld customisable
 - Second Life (utilisé pour des études médicales, sociologiques et pour l'enseignement)
 - OpenGrid (version Open Source des serveurs SL)

Conclusion

- Intergiciels généralistes
 - Peu adaptés sauf pour des applications avec peu d'utilisateurs (car trop « fiables » et trop « génériques »)
 - Corba 3 avec ses extensions TR ou ICE pourraient être plus intéressants
- Intergiciels spécifiques
 - Plus ils sont de haut niveau, plus ils sont spécifiques à un type d'application (FPS ou MMO)
 - Peu utilisés pour les jeux à gros succès (sauf Unreal Engine)
 - Très intéressants pour s'entraîner (développer son premier jeu en réseau), pour les indépendants ou pour la recherche (en IA, en 3D, en psycho, socio...)