Middlewares for Networked VEs

Patrice Torguet

torguet@irit.fr

Paul Sabatier University/IRIT/
VORTEX

Schedule

- Introduction
- Generic Middlewares
  - RPC, RMI, CORBA
- Specific Middlewares
  - Direct Play, TorqueNL, Raknet, Unreal net code
  - HLA, Openskies, Vega Prime, Delta 3D
  - Multiverse, Virtools Multiuser Server, NeL
  - BigWorld, Hero Engine
- Conclusions

Introduction

- Middleware: software layer that is located between applications and the operating system
- Like an OS it offers services that many applications use
- Like an application, it is not required by every application and it uses OS services

Introduction

- Several Middlewares exist (DBMS, distibution, GUI, 3D...)
- Here: network related middlewares
- Generally speaking: they use sockets (and sometimes threads and synchronization tools) to offer higher level services

Introduction

- Service examples
  - Remote procedure/method calls
  - Naming/Location/Discovering
  - Migration/Load balancing
  - Message/events/notifications transfer

RPC : Remote Procedure Call

- Client/server based
- Remote procedure call between a client and a server
  - Client calls a local procedure (called stub or proxy)
  - The local procedure uses a socket connexion to send a procedure identifier and parameters to the server (request)
  - The server receives the request using a socket, extracts the procedure id and parameters (this is done by what is called a skeletton)
  - The skeletton runs the remote procedure and sends back results through the socket connection
- rpcgen pre-compiler
  - Generates stub and skeletton using a file that describes the procedure(s) interface(s) using a computer language independent syntax (RPCL: RPC Language)

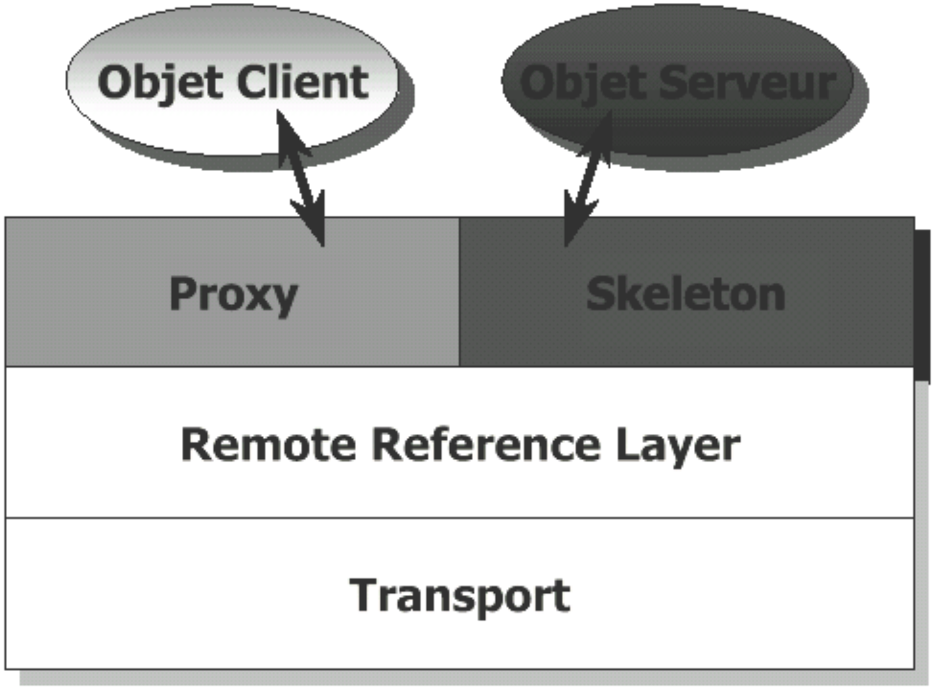
RPC : Remote Procedure Call

- Presentation layer called XDR (eXternal Data Representation)
  - Network way of presenting primitive as well as structured data (arrays, sequences, structures...)
  - Independant of
    - Computer architecture (little endian/big endian)
    - Computer language (e.g. manages line/collumn order in C and Fortran arrays)
    - OS specifics (ASCII, EBCDIC)
- Limitations:
  - No object oriented concepts (encapsulation, inheritence, polymorphism)
  - No advanced services: naming…
- Successors:
  - RMI : mono language, multi platform
  - CORBA : multi language, multi platform
  - COM : multi language, mono platform (multi for DCOM)
  - SOAP / .NET / web services : multi language, multi platform, web oriented

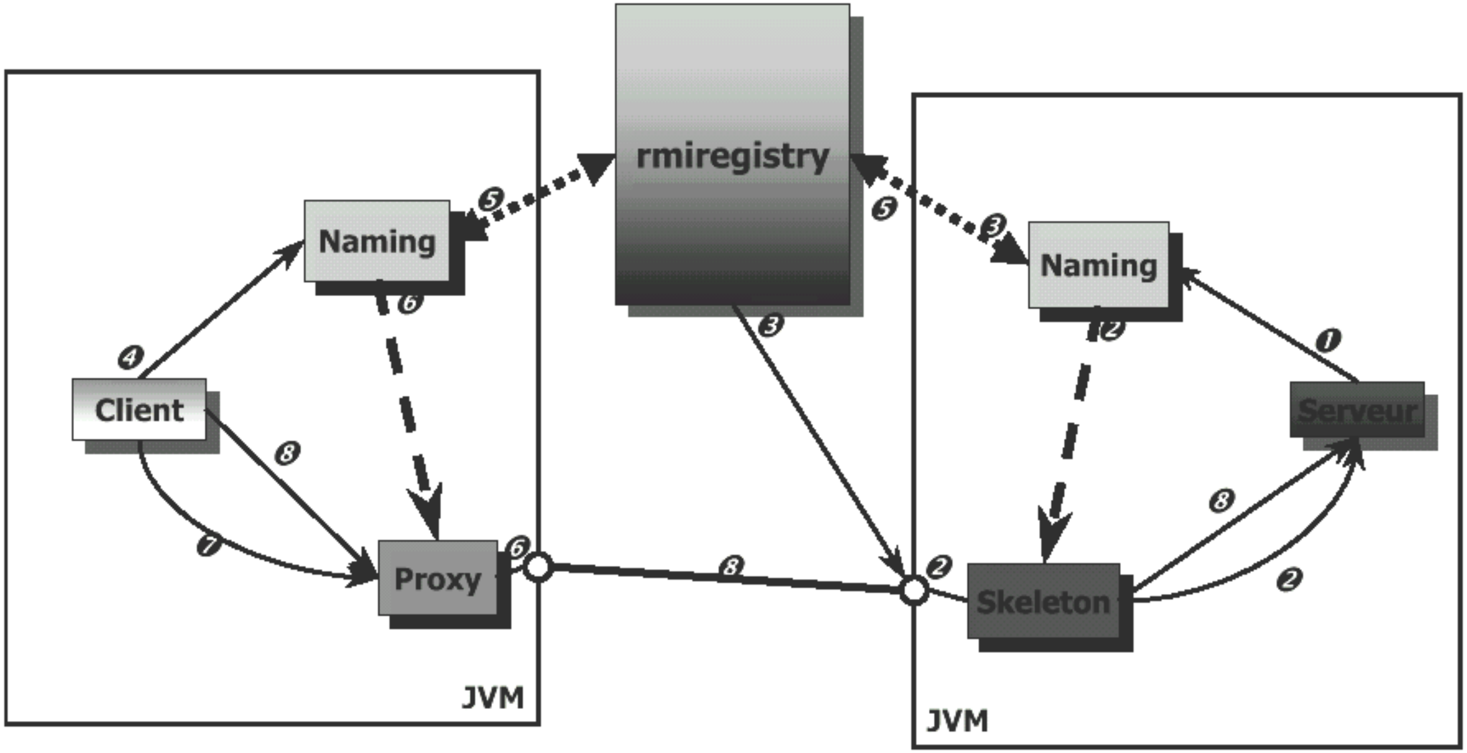
RMI: Remote Method Invocation

- Java based RPC
  - Invocation of methods on distributed objects
  - Uses all the language specifics => very simple to implement
- Tools
  - Pre-compiler for stubs/skelettons: rmic
  - Simple naming service: rmiregistry (you need to specify the server host)
  - Activation: rmid
- Mono language and multi platform: from JVM to JVM
- Object oriented: uses serialization
- Dynamic: download class files for stubs and parameters through HTTP (http://) or NFS/SMB (file:/)
- Somewhat secured: SecurityManager and .policy file

RMI: Remote Method Invocation



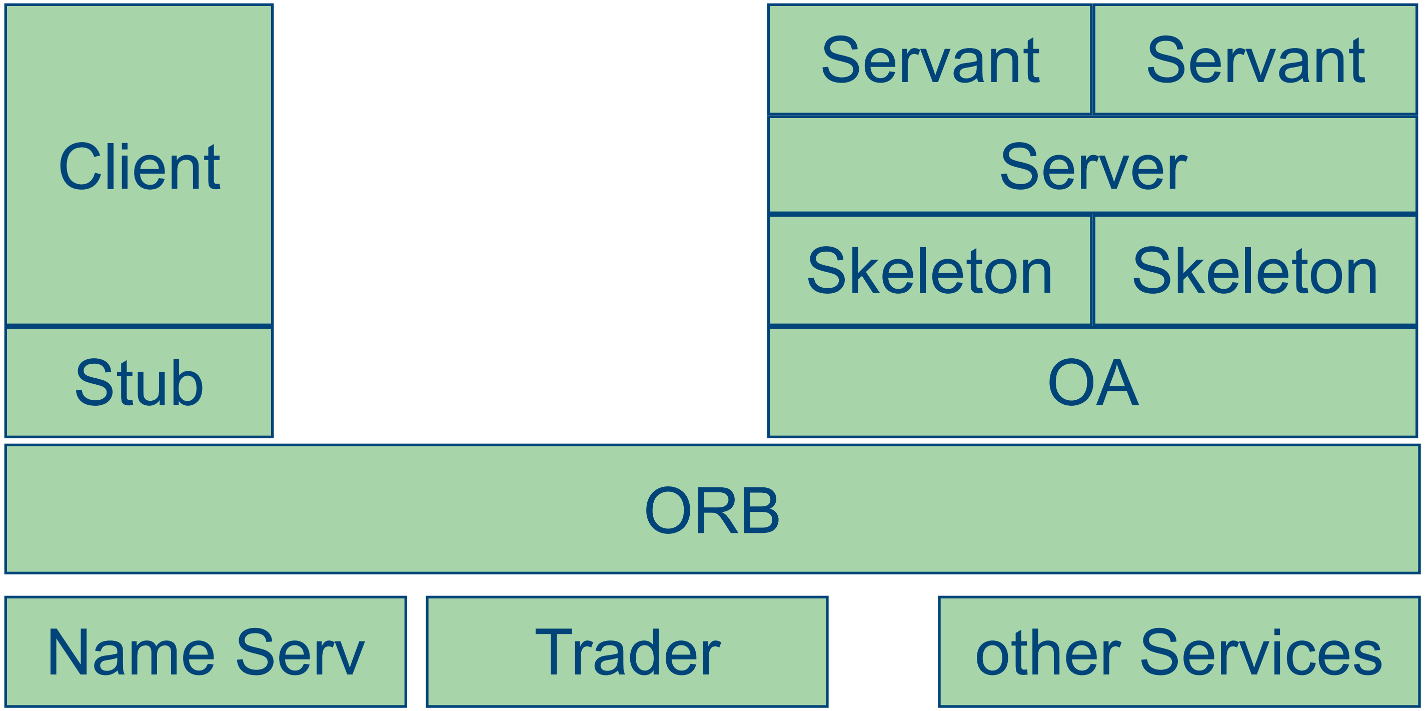
RMI: Remote Method Invocation



CORBA: Common Object Request Broker Architecture

- Object oriented RPC + several services
  - Language independent => uses IDL (Interface Definition Language)
- Tools
  - IDL pre-compiler generates stubs and skeletons
  - Advanced Naming service, trading service (yellow pages)
  - Manages events, persistence, realtime, streaming...
- Multi-language and multi-platform
- Interoperability between ORBs: GIOP/IIOP

CORBA : Architecture (static only)



Client

Stub

Servant

Servant

Server

Skeleton

Skeleton

OA

ORB

Name Serv

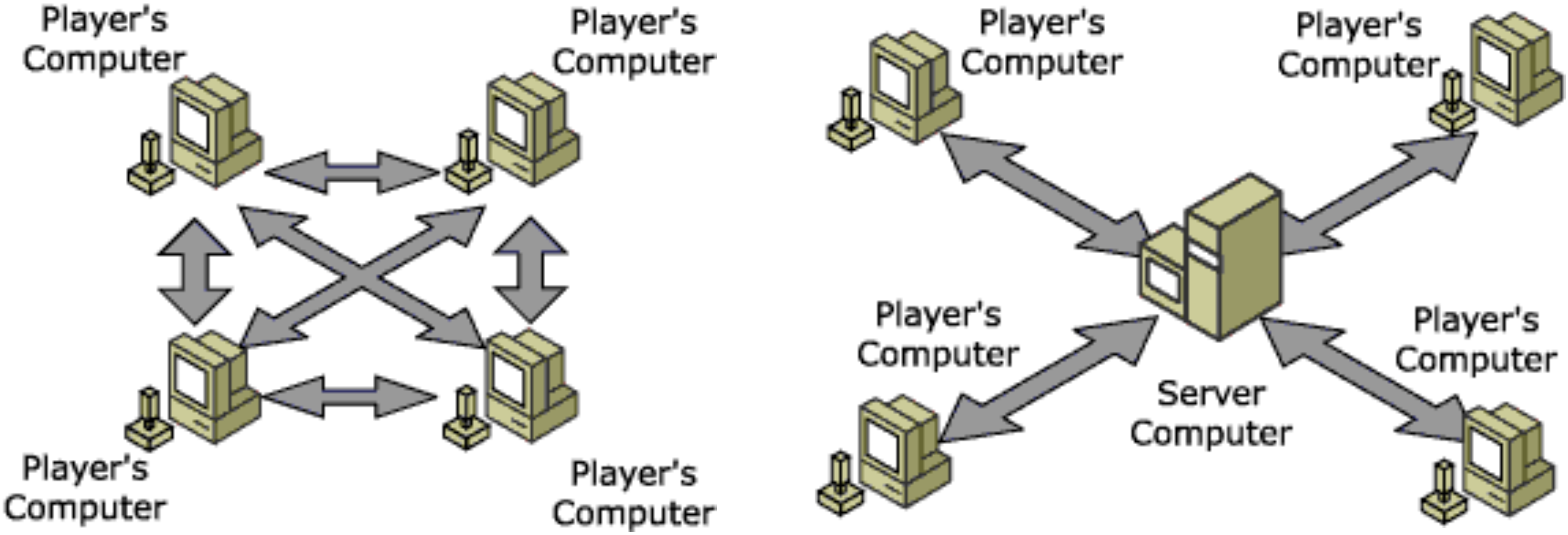
Trader

other Services

Direct Play

- Was in Direct X (3 – 9 - abandoned for 10 replaced by XBox Live)
- Supported 2 architectures
  - Totally connected peer to peer
  - Client/Server
- Server manages group filtering

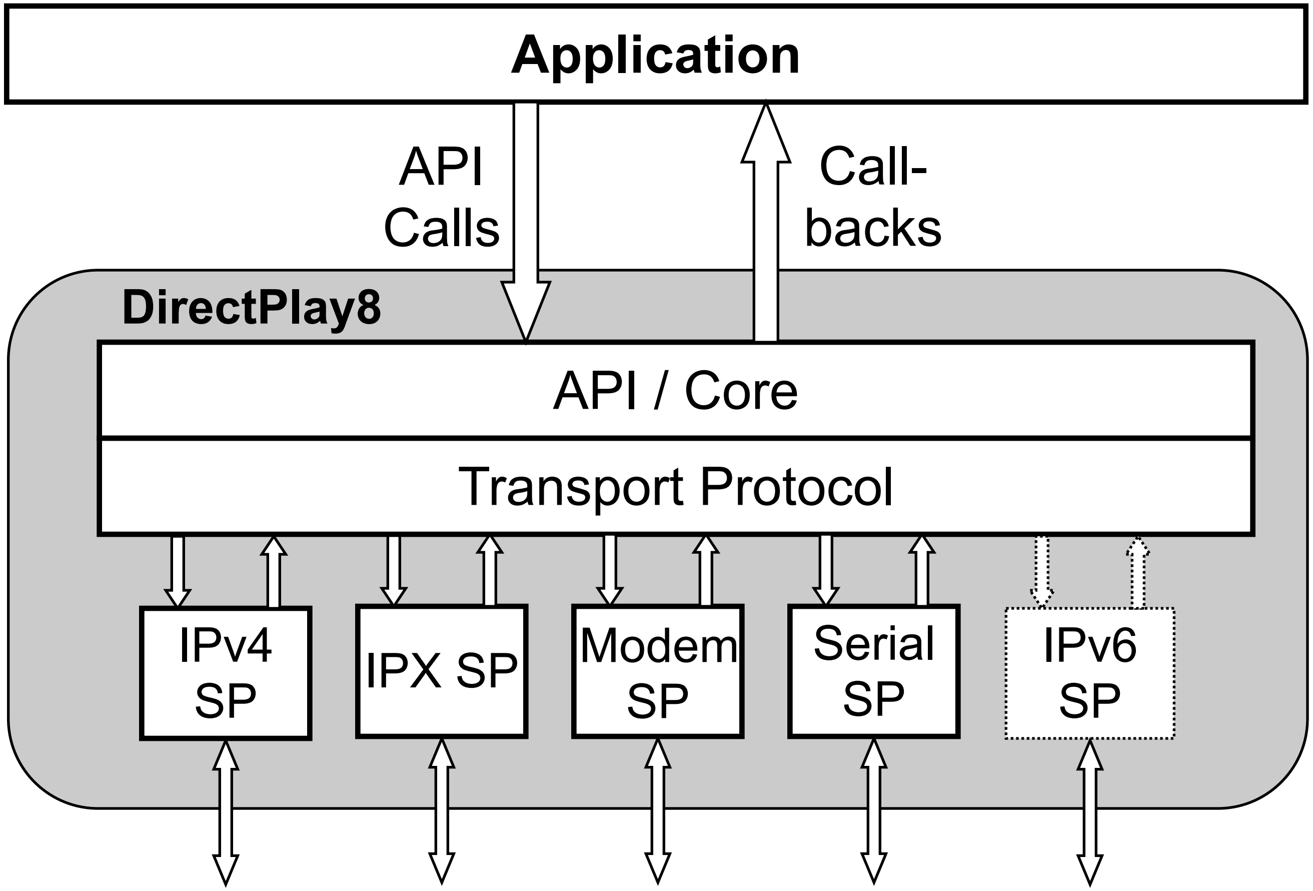
Direct Play : architectures



Direct Play

- Multiuser audio chat
- Interesting testing and validation tools
  - Netmon (realtime monitoring)
  - Dp8Sim (network simulator - limits bw adds network losses)

Direct Play: layers architecture



TorqueNL

- From “Starsiege: TRIBES” and TRIBES 2 games
- FPS oriented Client/Server Architecture
  - Manage data compression
    - Bit wise (booleans, x-bits integers, several precisions for floating point values)
  - Several QoSes
    - Ordered and reliable
    - Reliable only
    - Best effort
    - Up to date resends (if it needs to resend messages it will send them with updated date)
    - As fast as possible (data are resent until acknowledged)

TorqueNL

- Other features:
  - Interpolation/Extrapolation and Dead-Reckoning
  - Client specific filtering (application can tell if an object is visible or non visible for a client)
  - Priorities for object updates
  - Static data download at startup
  - RPC
  - Symmetric and Asymmetric Ciphering
  - Firewall/NAT crossing
  - Meta-server for finding servers
- C++ either GPL or commercial license
- http://www.opentnl.org/

Raknet

- Client/Server or Peer to Peer with App Level MCast
- Multi-platform (Windows, Linux, Consoles)
- Features:
  - Message compression
  - Several QoSes (reliable, sequenced, ordered)
  - Ciphering
  - Auto patcher
  - Voice chat
  - Serialization

Raknet

  - RPC
  - network emulation (latency, limited bandwidth)
  - Meta-server
  - Firewall/NAT crossing
- C++ - Development license is free + commercial deployment license
- http://www.rakkarsoft.com/

Unreal net code

- Unreal engine networking layer
- Client/server for FPS
  - Object replication with priorities
  - Several QoSes
  - Non visible object filtering (linked to BSP trees)
  - RPC
  - Quantification for data compression
  - Dead-reckoning
- Can be used with UnrealScript (free for mods)
- Unreal engine Commercial license for C++ source code
- http://unreal.epicgames.com/Network.htm

HLA (High Level Architecture)

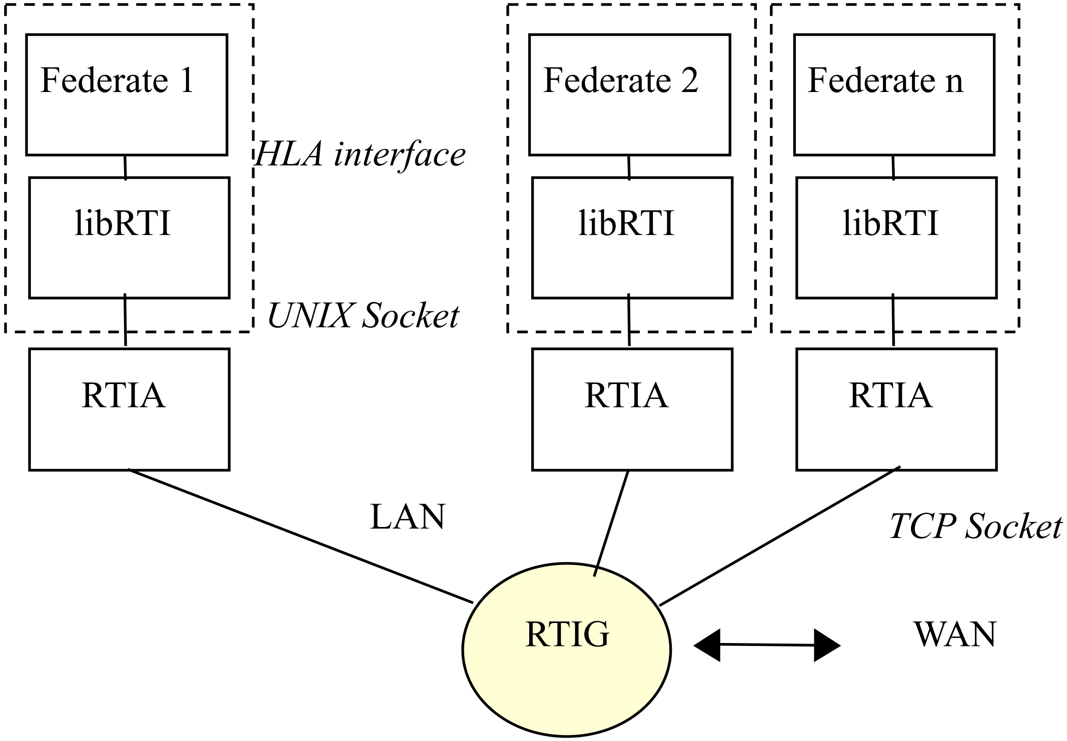
- US DoD
- Motivations
  - Allow interoperability and reusability of existing and future simulations
  - Specifies several APIs and rules that can be implemented in several ways
  - Tries to solve DIS problems
- Availability
  - Documentation/specs: http://hla.dmso.mil/
  - Opensource version: http://savannah.nongnu.org/projects/certi/

HLA (High Level Architecture)

- Composed of:
  - Federation rules: general principles that should be applied to distributed simulation development in order to be HLA compatible
  - RTI (Run-Time Infrastructure): middleware which manages the distributed simulation
  - OMT (Object Modeling Template) : generic object oriented model used to define simulation objects and events

HLA (High Level Architecture)

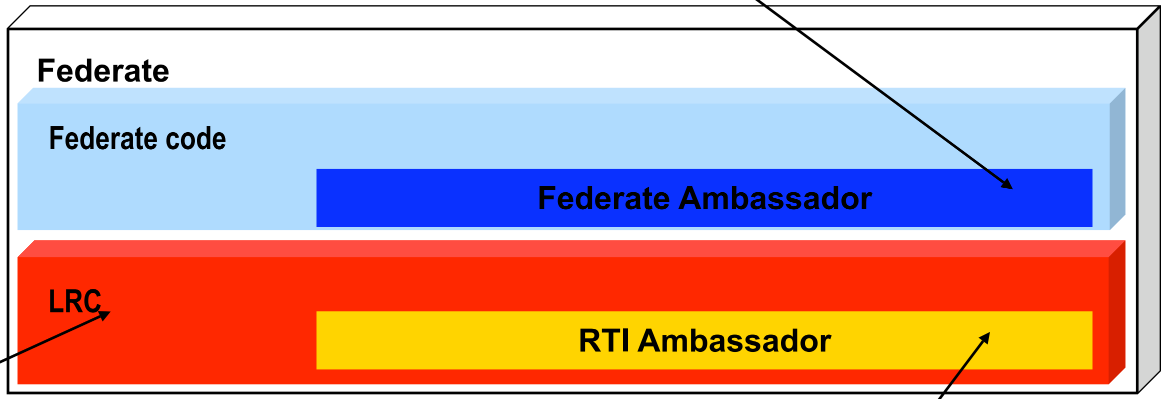
- Architecture example (CERTI)



HLA (High Level Architecture)

- libRTI
  - Offers services to simulators (called federates)
  - Federates communicate with each other through libRTI calls
  - C++ (MaKRTI, CERTI), Java (pitchRTI)...

Federates must define the abstract class called FederateAmbassador



Federate

Federate code

Federate Ambassador

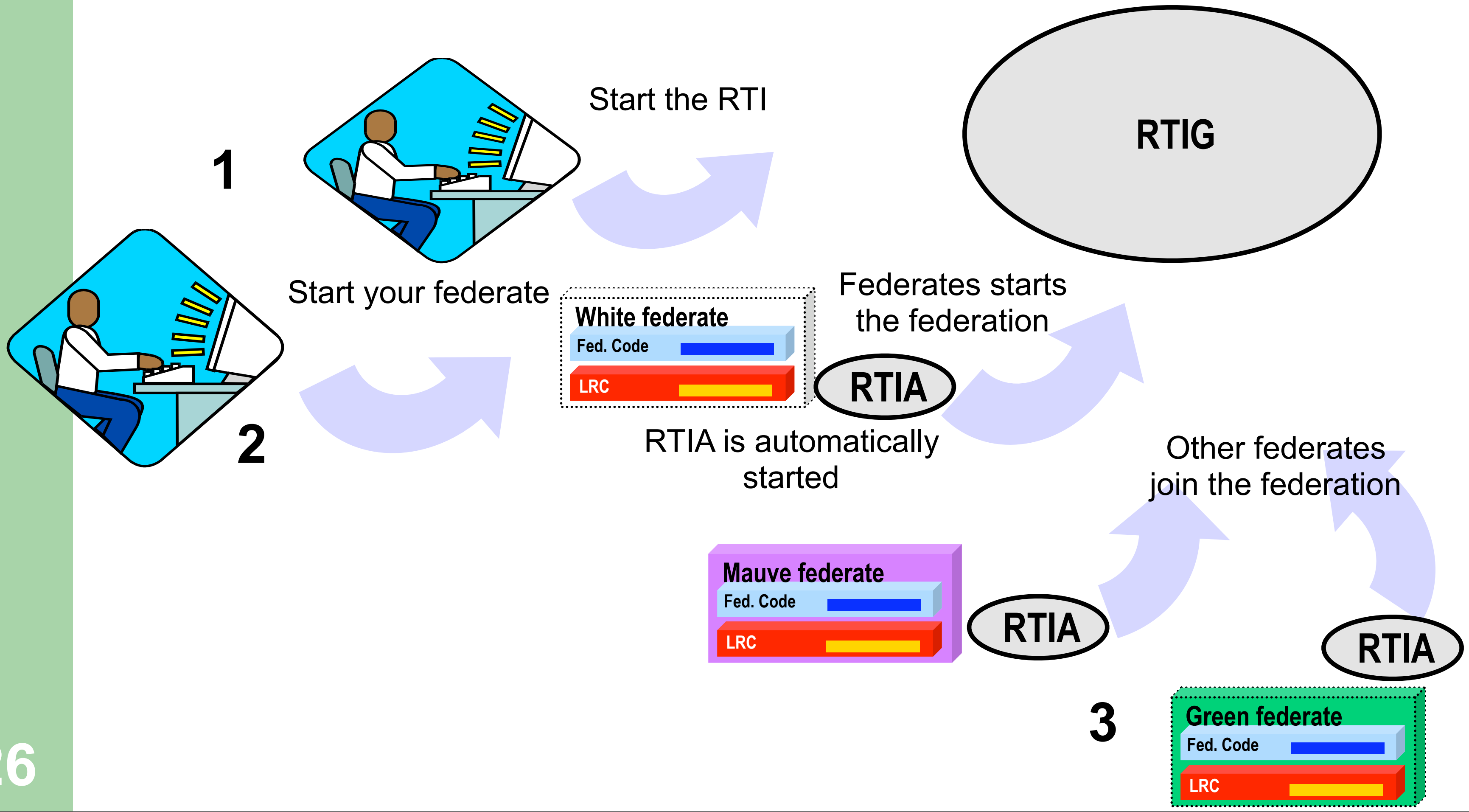
LRC

RTI Ambassador

Local RTI components (LRC) offer services as specified in the IFSpec

LRC main class is called RTIAmbassador

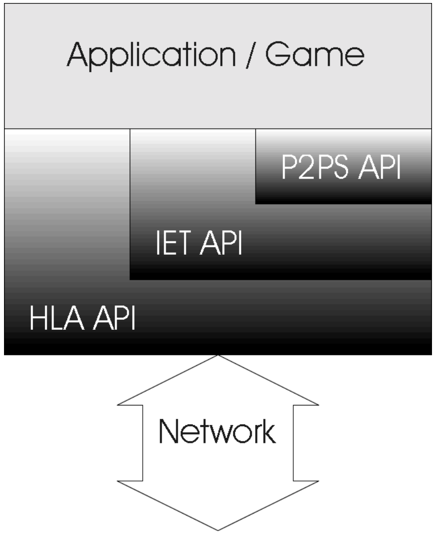
HLA (High Level Architecture)



HLA (High Level Architecture)

- Other (commercial) RTIs
  - pRTI from pitch.se
  - Mäk RTI from Mäk (mak.com)
  - Openskies (next slides)

Openskies

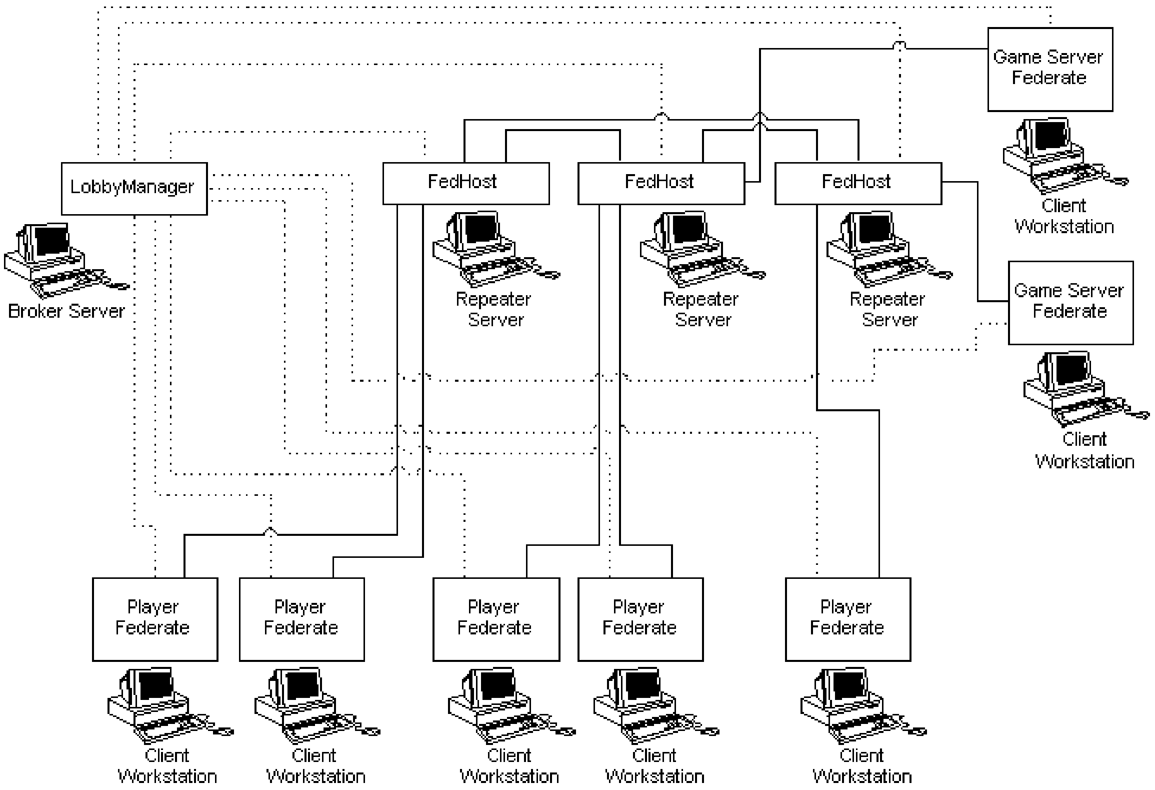


- Several APIs
  - low level: HLA (RTI)
  - IET (Import Export Table) HLA code generator (manages duplicated objects updates)
  - P2PS (Point to Point Switch Messaging Interface) allows objects to send messages to other objects on any computer and even if they are not currently created (buffered messages)

Openskies

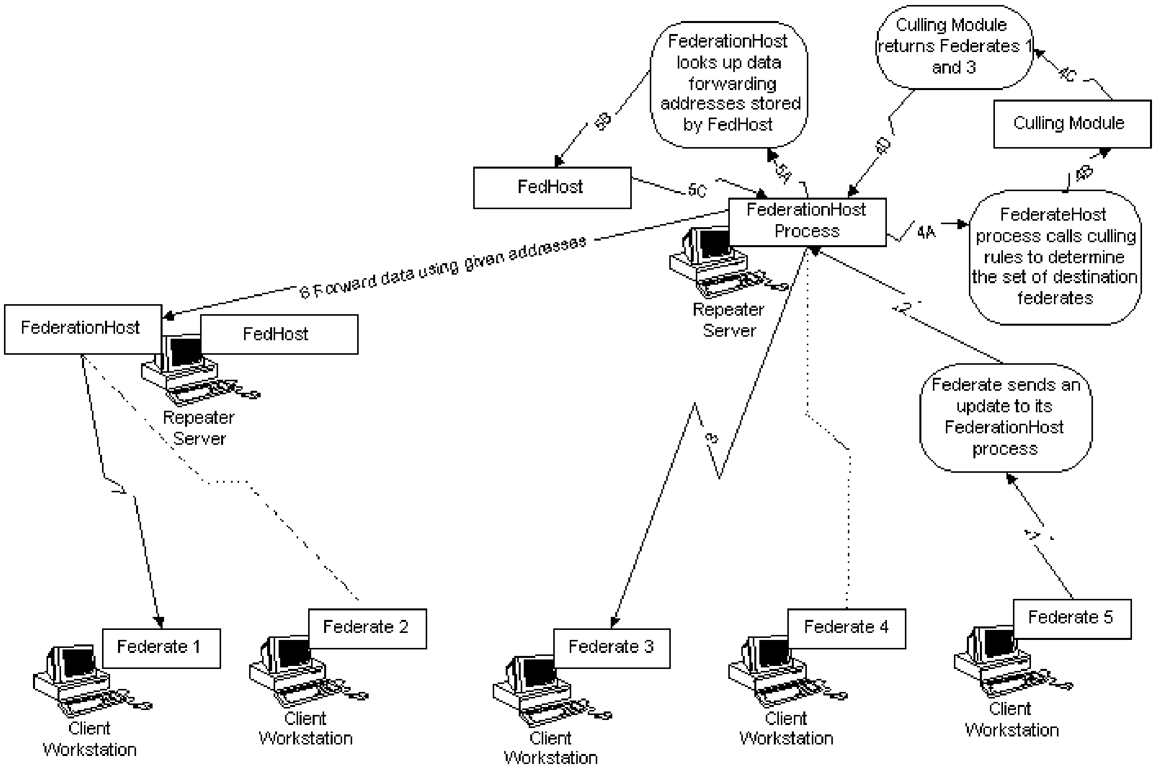
- Architectures
  - Peer to peer or client/multi servers
  - Dedicated Linux servers (specific Linux distribution)
  - 3 types of servers
    - LobbyManager (starting point - authenticates and directs to a FedHost)
    - FedHost (routes messages between federates)
      - Each can manage up to 500 federates
    - Game Server (manages the games specifics: AI...)
      - One or several
      - Coded like federates

Openskies


Game Server Federate
LobbyManager
FedHost
FedHost
FedHost
Client Workstation
Broker Server
Repeater Server
Repeater Server
Repeater Server
Game Server Federate
Client Workstation
Player Federate
Player Federate
Player Federate
Player Federate
Player Federate
Client Workstation
Client Workstation
Client Workstation
Client Workstation
Client Workstation

Openskies

- FedHost API allows the development of specific filters (culling)
  - Can use a 2D grid to filter on position



FederationHost looks up data forwarding addresses stored by FedHost

Culling Module returns Federates 1 and 3

4C

Culling Module

5B

4D

FedHost

5C

5A

FederationHost Process

4A

FederateHost process calls culling rules to determine the set of destination federates

4B

6 Forward data using given addresses

Repeater Server

2

FederationHost

FedHost

Repeater Server

Federate sends an update to its FederationHost process

3

1

Federate 1

Client Workstation

Federate 2

Client Workstation

Federate 3

Client Workstation

Federate 4

Client Workstation

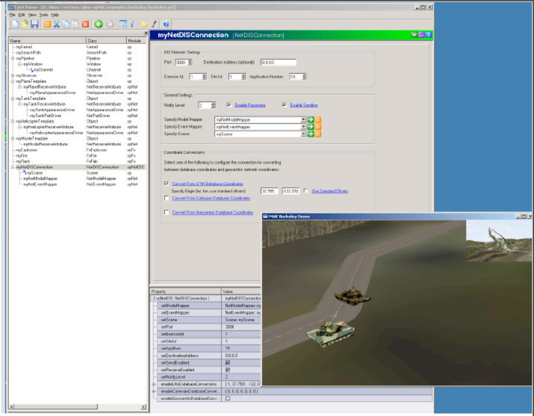
Federate 5

Client Workstation

Openskies

- C++ - Windows for federates
- Free dev license, commercial deployment license
- http://www.openskies.net/

Vega Prime (HLA/DIS mod)


LynX Prime - [D:/Alme/cvs/trees/alme-vpNet/examples/berkeley/berkeley.acf]
File Edit View Tools Help
MÄK Berkeley Demo

Name	Class	Module
myKernel	Kernel	vp
mySearchPath	SearchPath	vp
myPipeline	Pipeline	vp
myWindow	Window	vp
myChannel	Channel	vp
myObserver	Observer	vp
myPlaneTemplate	Object	vp
myPlaneReceiveAttribute	NetReceiveAttribute	vpNet
myPlaneAppearanceDriver	NetAppearanceDriver	vpNet
myTankTemplate	Object	vp
myTankReceiveAttribute	NetReceiveAttribute	vpNet
myTankAppearanceDriver	NetAppearanceDriver	vpNet
myTankPartDriver	NetPartDriver	vpNet
myHelicopterTemplate	Object	vp
myHelicopterReceiveAttribute	NetReceiveAttribute	vpNet
myHelicopterAppearanceDriver	NetAppearanceDriver	vpNet
myMissileTemplate	Object	vp
myMissileReceiveAttribute	NetReceiveAttribute	vpNet
myExplosion	FxExplosion	vpFx
myFire	FxFire	vpFx
myFlack	FxFlak	vpFx
myNetDISConnection	NetDISConnection	vpNetDIS
myScene	Scene	vp
myNetModelMapper	NetModelMapper	vpNet
myNetEventMapper	NetEventMapper	vpNet

myNetDISConnection (NetDISConnection)

DIS Network Settings

Port: 3000 Destination Address (optional): 0.0.0.0

Exercise Id: 1 Site Id: 1 Application Number: 15

General Settings

Notify Level: 2 ☑ Enable Receiving ☑ Enable Sending

Specify Model Mapper: myNetModelMapper

Specify Event Mapper: myNetEventMapper

Specify Scene: myScene

Coordinate Conversions

Select one of the following to configure the connection for converting between database coordinates and geocentric network coordinates:

☑ Convert From UTM Database Coordinates

Specify Origin (lat, lon, use standard offsets): 37.755 -122.370 □ Use Standard Offsets

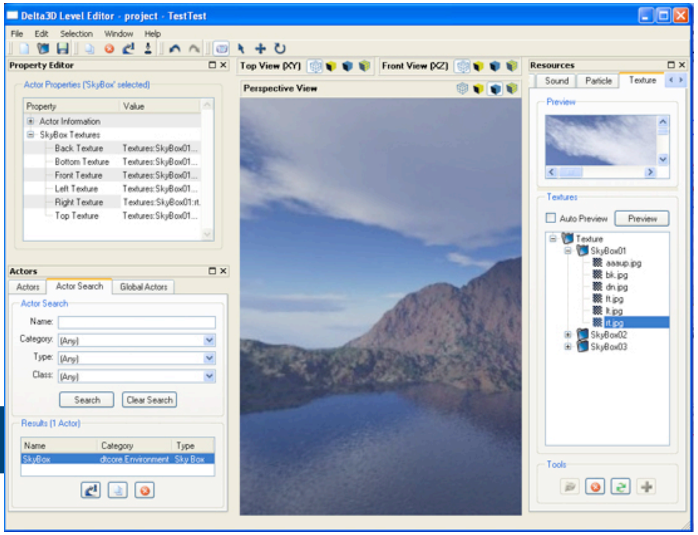
□ Convert From Cartesian Database Coordinates

□ Convert From Geocentric Database Coordinates

Property	Value
[vpNetDIS : NetDISConnection]	myNetDISConnectio
setModelMapper	NetModelMapper: my
setEventMapper	NetEventMapper: my
setScene	Scene: myScene
setPort	3000
setExerciseId	1
setSiteId	1
setAppNum	15
setDestinationAddress	0.0.0.0
setSendEnabled	☑
setReceiveEnabled	☑
setNotifyLevel	2
enableUtmDatabaseConversions	[1, 37.7551, -122.37
enableCartesianDatabaseConver...	[0, 0.0, 0.0, 0.0]
enableGeocentricDatabaseConv...	□

- Vega Prime
  - GUI app used to create realtime 3D simulators without coding
  - Modular architecture (marine, atmospheric effects, infrared rendering, radar...)
- HLA/DIS module
  - Distributes the simulation using HLA and the RPR FOM (DIS equivalent for HLA) or DIS
  - No need to code
  - Based on Mâk VR-Link (simulation distribution platform using DIS, HLA or TENA)
- http://www.presagis.com/products/multigen_paradigm/details/vegaprime/

Delta 3D



Property	Value
Actor Information	
SkyBox Textures	
Back Texture	Textures:SkyBox01...
Bottom Texture	Textures:SkyBox01...
Front Texture	Textures:SkyBox01...
Left Texture	Textures:SkyBox01...
Right Texture	Textures:SkyBox01:rt.
Top Texture	Textures:SkyBox01...

Name:
Category: (Any)
Type: (Any)
Class: (Any)

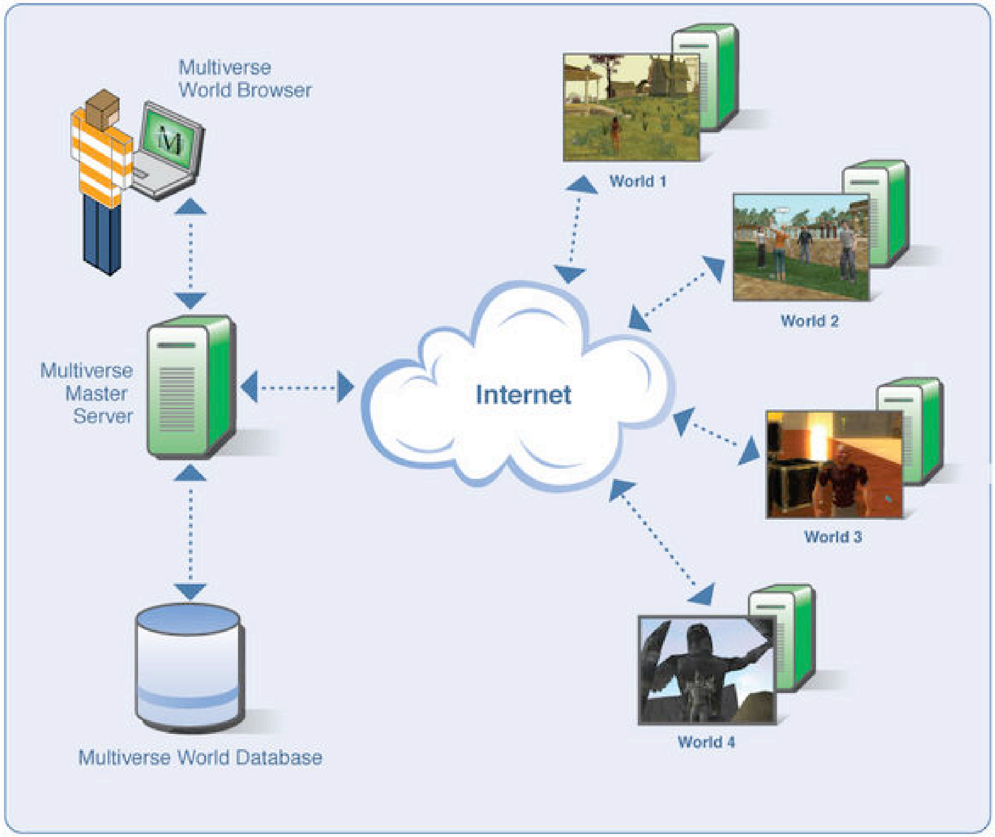
Name	Category	Type
SkyBox	dtcore.Environment	Sky Box

- Opensource “Clone” of Vega Prime
- Based on several toolkits
  - Open Scene Graph (3D)
  - Open Dynamics Engine (ODE - physics engine)
  - Open AL (audio)
- High level classes
  - Specific network Toolkit
  - Encapsulates HLA (using a difficult to find RTI but can (?) be ported to CERTI)
- GUI app for creating realtime 3D simulations

Multiverse

- Client/Server(s) complete environment targeted for MMORPGs
  - Manages quests, skills, combat, communication, inventory and trading, crafting, game economy, AI, billing...
  - Virtual world building tools
  - World Browser (used to locate worlds)
- APIs for client and server customization
- Commercial license (10% of the game’s earning)
- http://multiverse.net/

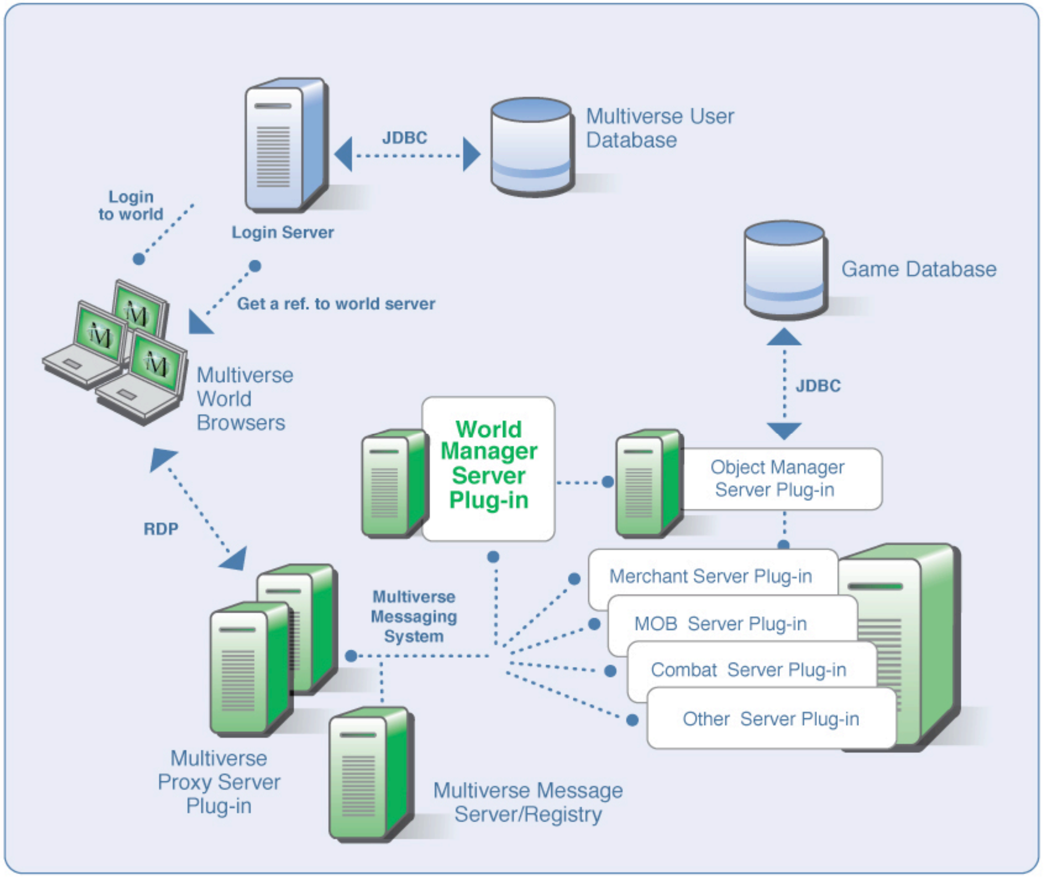
Multiverse


Multiverse World Browser
World 1
World 2
Multiverse Master Server
Internet
World 3
World 4
Multiverse World Database

Multiverse

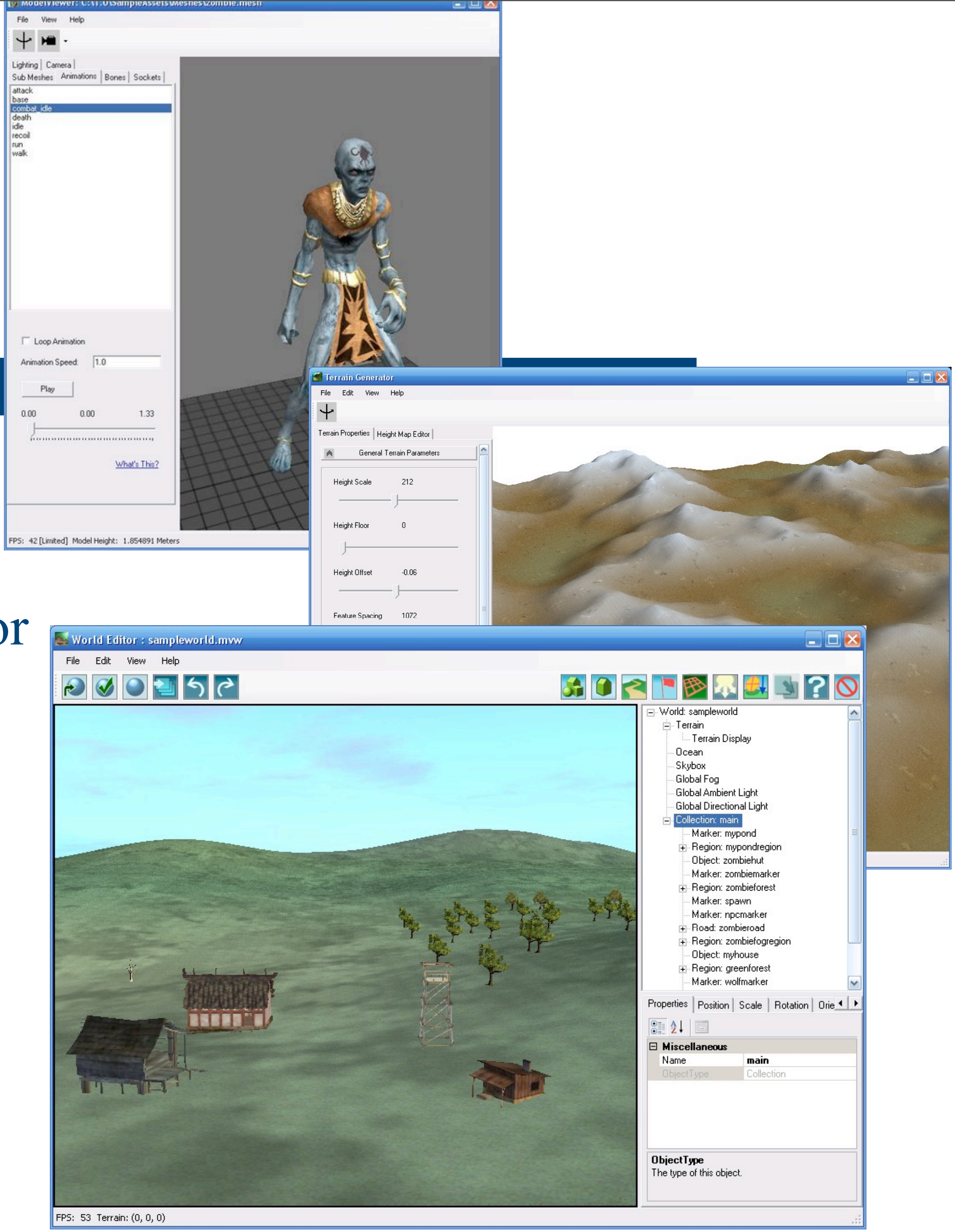
- Client is written in C++ with Ogre3D and can be customized using Python (GUI as well)
- Server is written in Java
- JDBC for persistence
- Plug-in architecture
  - standard plug-ins (all are optional)
  - API used to code specific plug-ins

Multiverse


Login Server
JDBC
Multiverse User Database
Login to world
Get a ref. to world server
Game Database
JDBC
Multiverse World Browsers
World Manager Server Plug-in
Object Manager Server Plug-in
RDP
Merchant Server Plug-in
Multiverse Messaging System
MOB Server Plug-in
Combat Server Plug-in
Other Server Plug-in
Multiverse Proxy Server Plug-in
Multiverse Message Server/Registry

Multiverse

- Tools
  - ModelViewer
  - Terrain Generator
  - WorldBuilder



ModelViewer: C:\...\SampleAssets\Meshes\zombie.mesh

File View Help

Lighting | Camera | Sub Meshes | Animations | Bones | Sockets

attack
base
combat_idle
death
idle
recoil
run
walk

☐ Loop Animation

Animation Speed: 1.0

Play

0.00 0.00 1.33

What's This?

FPS: 42 [Limited] Model Height: 1.854891 Meters

Terrain Generator

File Edit View Help

Terrain Properties | Height Map Editor

General Terrain Parameters

Height Scale 212

Height Floor 0

Height Offset -0.06

Feature Spacing 1072

World Editor : sampleworld.mvw

File Edit View Help

World: sampleworld
- Terrain
  - Terrain Display
- Ocean
- Skybox
- Global Fog
- Global Ambient Light
- Global Directional Light
- Collection: main
  - Marker: mypond
  - Region: mypondregion
  - Object: zombiehut
  - Marker: zombiemarker
  - Region: zombieforest
  - Marker: spawn
  - Marker: npcmarker
  - Road: zombieroad
  - Region: zombiefogregion
  - Object: myhouse
  - Region: greenforest
  - Marker: wolfmarker

Properties | Position | Scale | Rotation | Orie

Miscellaneous

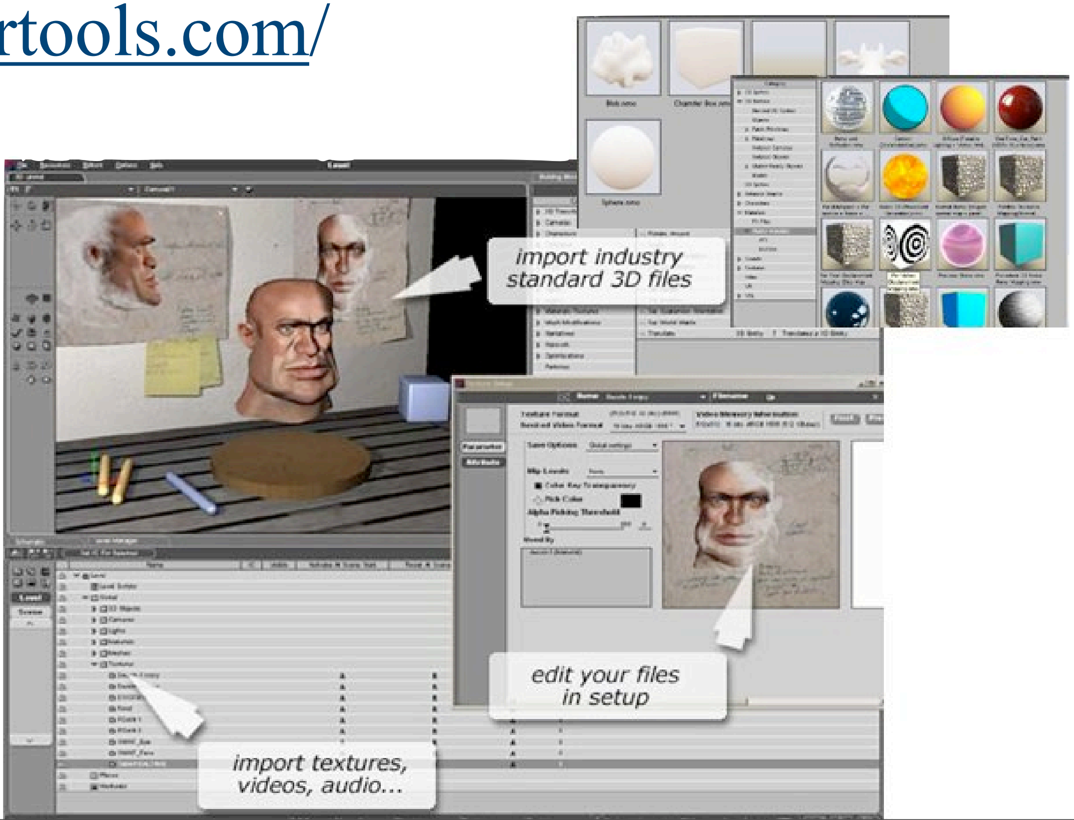
Name	main
ObjectType	Collection

ObjectType
The type of this object.

FPS: 53 Terrain: (0, 0, 0)

Virtools Multiuser Server

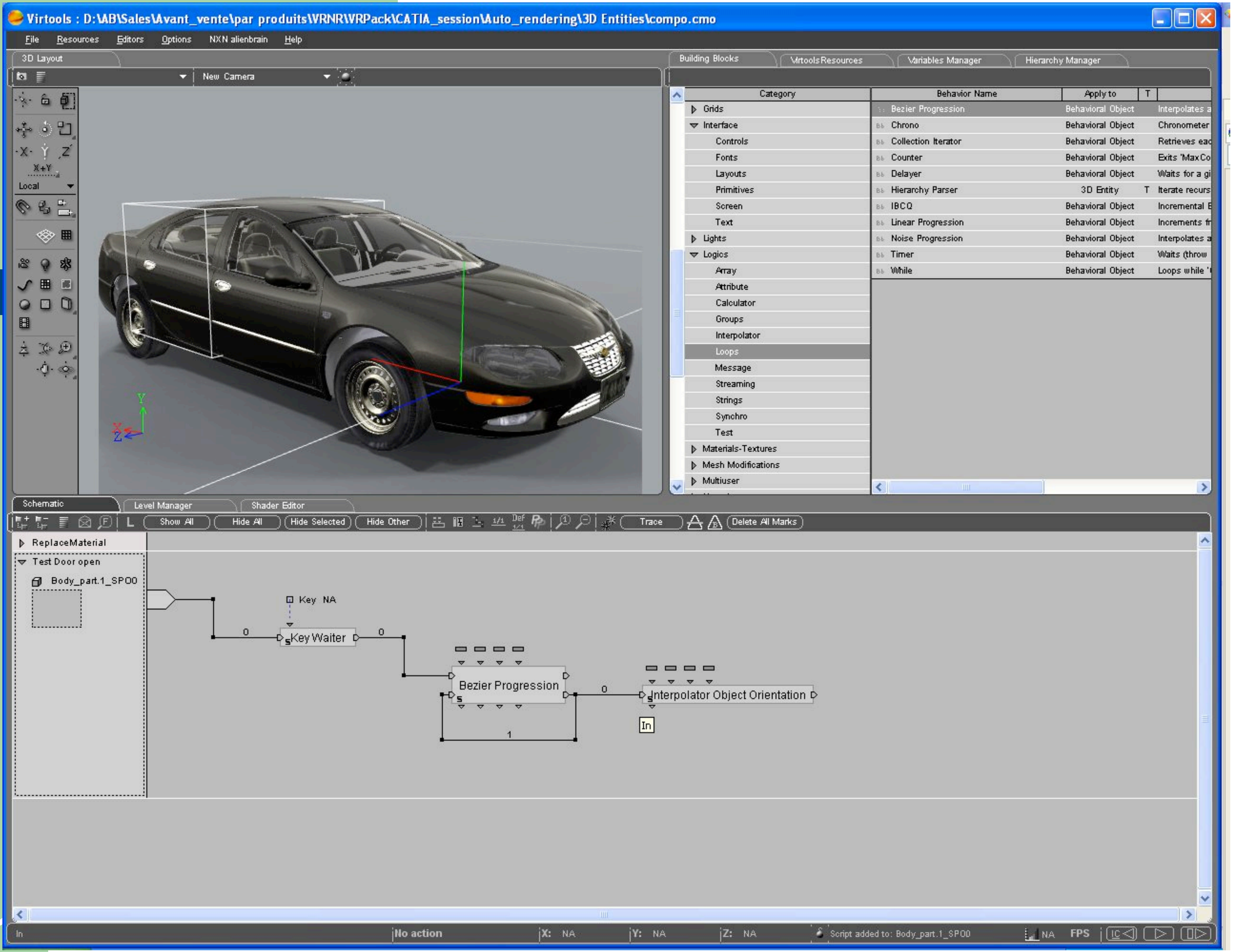
- Virtools is a realtime 3D application generator
- http://www.virtools.com/



import industry standard 3D files

edit your files in setup

import textures, videos, audio…



Virtools : D:\AB\Sales\Avant_vente\par produits\VRNR\VRPack\CATIA_session\Auto_rendering\3D Entities\compo.cmo

Behavior Name	Apply to	T	
Bezier Progression	Behavioral Object		Interpolates a
Chrono	Behavioral Object		Chronometer
Collection Iterator	Behavioral Object		Retrieves eac
Counter	Behavioral Object		Exits 'MaxCo
Delayer	Behavioral Object		Waits for a gi
Hierarchy Parser	3D Entity	T	Iterate recurs
IBCQ	Behavioral Object		Incremental B
Linear Progression	Behavioral Object		Increments fr
Noise Progression	Behavioral Object		Interpolates a
Timer	Behavioral Object		Waits (throw
While	Behavioral Object		Loops while '

In | No action | X: NA | Y: NA | Z: NA | Script added to: Body_part.1_SP00 | NA

Virtools Multiuser Server

- Features
  - 3D object importation
  - Animations management
  - GUI created scripts (connected building blocks using data flow)
  - SDK for building block or complete application development
  - Web plugin
- Multiuser Server
  - Manages distributed features
  - Peer to peer (one client manages some server features) or client/ server
  - Dead-reckoning
  - Not that many details

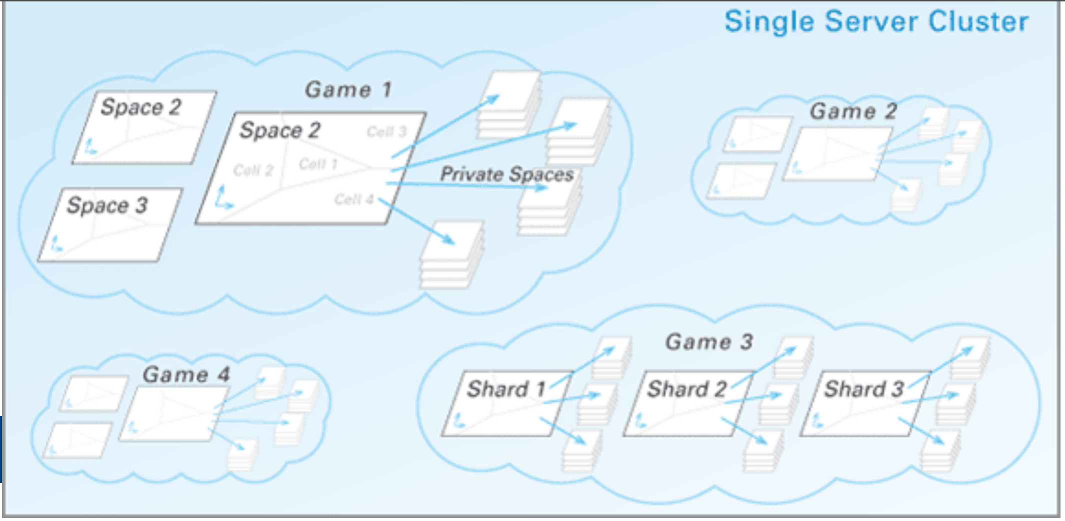
NeL

- MMO engine
- Developed for Ryzom (http://www.ryzom.com) and Ryzom Ring (scenario editor for Ryzom)
- Several GPL libraries
  - 3D, network, simple physics engine
- NelNet
  - Proposes a typical "sharded" architecture
  - Each shard must have an administration server and one or more game servers

NeL

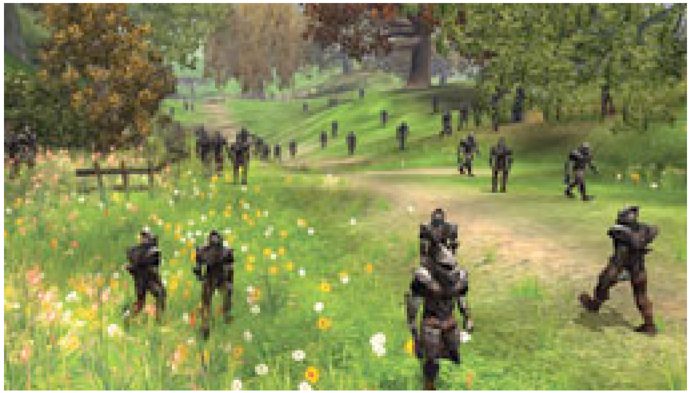
  - Most of the server code has to be developed
  - NelNet only offers low-level libraries
- C++ for Linux (server) and multi-platform (clients)
- http://www.nevrax.org/

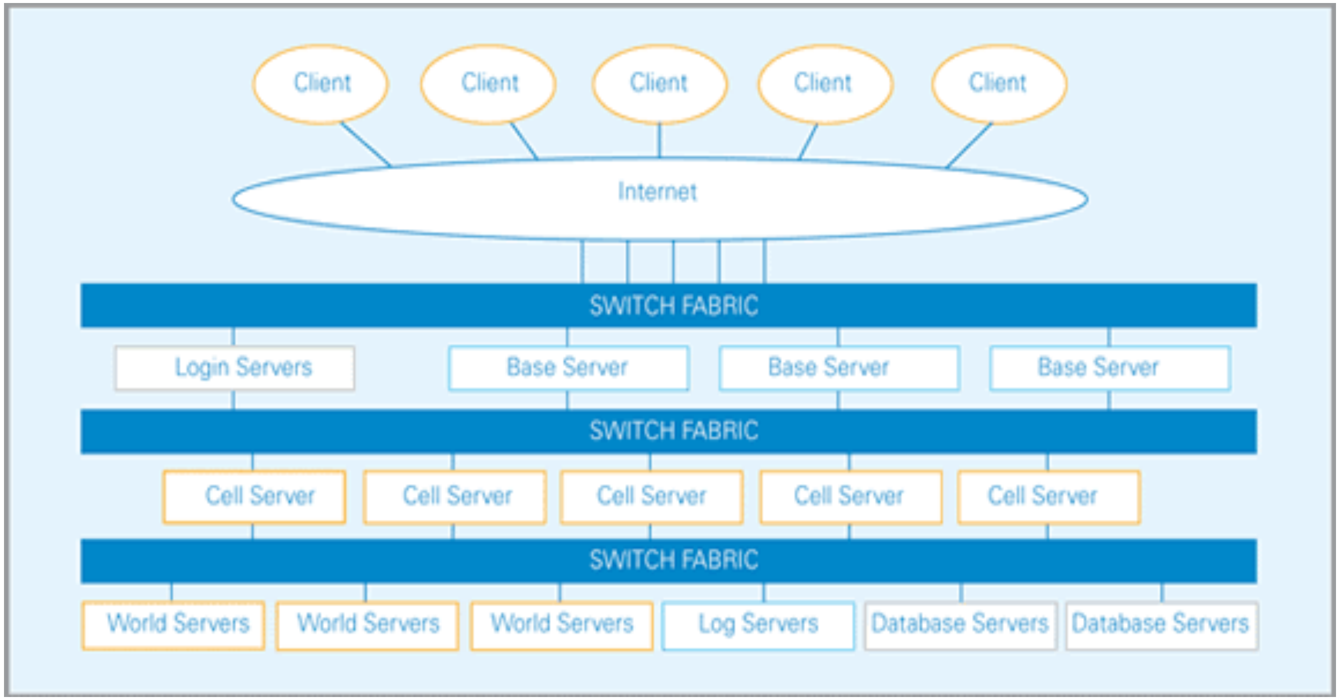
BigWorld



- Complete MMO engine
- Offers a complete MMO you can customize
- Licensed by several future MMOs (e.g. Stargate Worlds, Okuto no Ken MMO...)
- Server side
  - Dynamic load balancing
  - Dynamic seamless architecture
  - Fault tolerant
  - Can manage several instances, shards or even games on one cluster

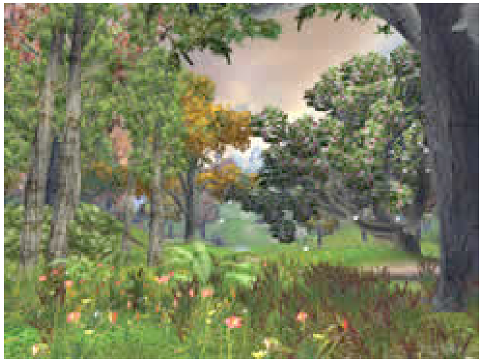
BigWorld





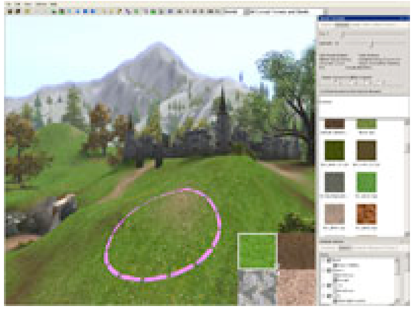
Client
Client
Client
Client
Client
Internet
SWITCH FABRIC
Login Servers
Base Server
Base Server
Base Server
SWITCH FABRIC
Cell Server
Cell Server
Cell Server
Cell Server
Cell Server
SWITCH FABRIC
World Servers
World Servers
World Servers
Log Servers
Database Servers
Database Servers

BigWorld



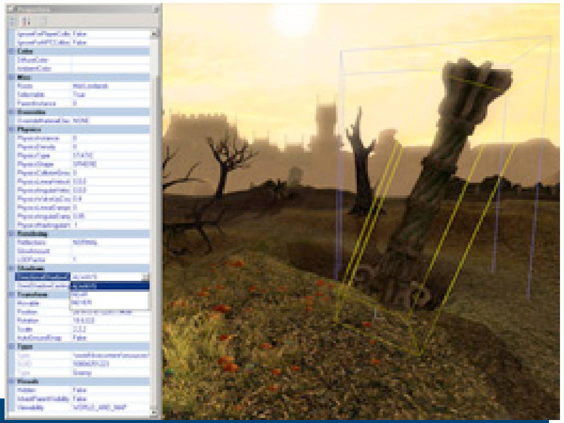
  - Manage dynamic adding/removal of servers in the cluster
  - Manages priorities and LOD (quantification ?) for messages for optimally managing the available bandwidth between the cluster and a client
  - Monitoring tools
  - C++ with Python customization
- Client side
  - Windows and consoles
  - Specific state of the art 3D engine

BigWorld



- Content production tools
  - World editor: terrain, flora, simplified building creation. Collaborative tool.
  - Model editor: imports models and allow animations editing
  - Particles effects editor
- Commercial license
- http://www.bigworldtech.com/

Hero Engine



- Similar to BigWorld
- Similar tools but seem more interactive and collaborative
- Very little details on the client/server architecture
- Licensed by Bioware for their future MMO
- http://www.play.net/playdotnet/platform/home.asp

Conclusion

- Generic Middlewares
  - Not really suited except for small scale applications (too much reliability)
  - Corba 3 and its extensions could become more interesting
- Specific Middlewares
  - The more high level they are the more specific (FPS or MMO) and expensive they are
  - Not really used for commercial games (Unreal Engine is an exception here)
  - For mangas/books/films licenses ?
  - Very interesting for developing your skills (for your first networked game) and for research (if you don’t plan to commercialize your research)