

Solipsis: A Decentralized Architecture for Virtual Environments

Davide Frey

Joint work with

E. Anceaume, A-M. Kermarrec

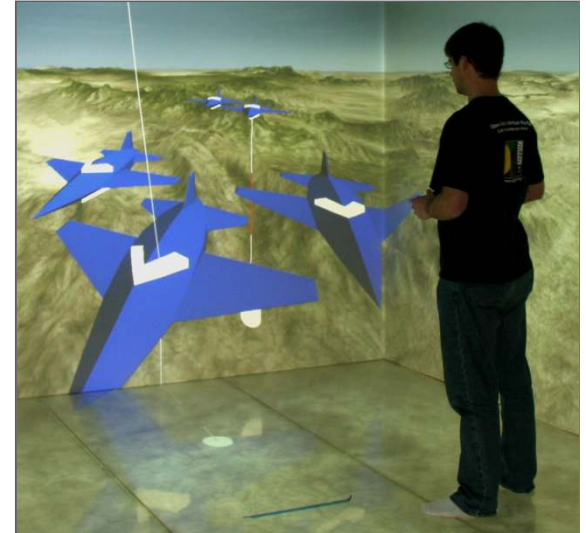
F. Le Fessant, R. Piegay, J. Royan



As Scalable As Possible

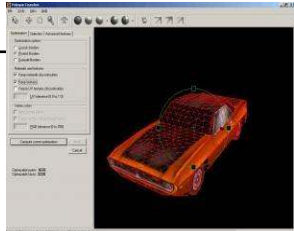



The (virtual) world as We Know it

- Simulation of space in which users can interact with each others and are embodied by virtual **projections of themselves**
- Common Assumptions
 - Scalability
 - Persistence
 - Massive Sharing



A Naïve Taxonomy



	stand-alone	<ul style="list-style-type: none">• Data / Content A single virtual object display of a single 3D object, 2D websites, or any multimedia contents	
	Centralized - Server(s)	<ul style="list-style-type: none">• MetaWorld (aka Virtual world) Perceived as single location space (world) Either single user or massive multi-user Example: Second Life, There, WOW	
	Group of Servers	<ul style="list-style-type: none">• MetaGalaxy A group of MetaWorlds More than likely massive multiuser Interconnected (not stand-alone) Example: Active Worlds	
	Decentralized - World scale infrastructure	<ul style="list-style-type: none">• Metaverse Multiple MetaGalaxy systems Linked within a perceived Virtual Universe, Although not existing on a central server	

Just What is a Metaverse

- A world scale infrastructure
- Vast amount of **virtual worlds** linked together
- Accessible via a single common user interface (browser)
- Not marking a break with the actual flat 2D web: but a smooth transition towards an immersive Internet making the most of both 2D and 3D



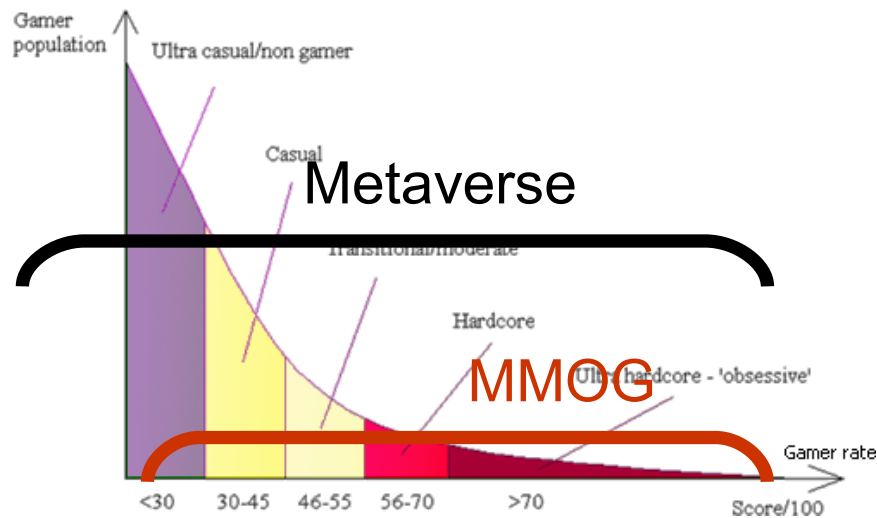
« Hiro is approaching the Street. It is the Broadway, the Champs Elysees of the Metaverse. It does not really exist. But right now, millions of people are walking up and down it »

**Neal Stephenson,
Snow Crash [p24]**

**Snow
Crash,
Neal
Stephenson
© 1993
Bantam
Books**

Just What is a Metaverse?

- Not an MMOG [massively multiplayer online game]
 - Tremendously bigger
 - No pre-existing contents nor a scenario to respect (UGC)
 - Maybe as many different business models as inner virtual worlds
 - For everyone, especially no gamer, and not hardcore gamers only

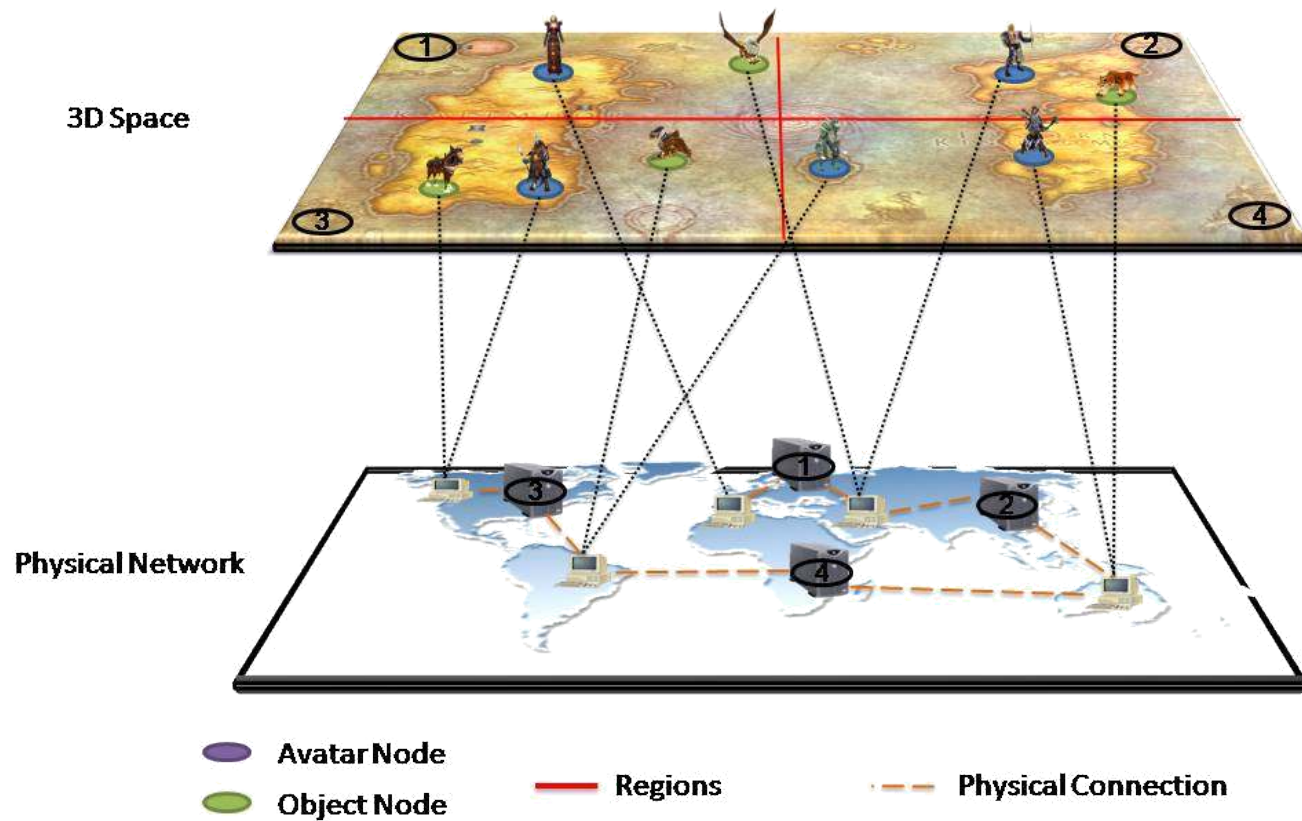


Metaverse Challenges

- Truly World-Wide Infrastructure
 - Support Multiple Independent Yet Interconnected Worlds
- Accessible in Variety of Situations
 - PC
 - PDA
 - Mobile Phones
- Requires Design of Highly Scalable Architectures
 - Distributed Access to Information
 - Computation of Physics

Existing VW architectures

- Generally based on region servers



Existing VW architectures

- Region servers responsible for synchronization:
 - Collision Detection
 - Physics animations
- Region servers are the only source for downloading scene models:
 - Require significant upload Bandwidth
- If servers are overloaded, audience is limited (50-70 clients on SL)
- No scalability for a Huge Metaverse (Just how many servers?)

The Solipsis Vision

- A P2P Architecture
 - Scalability to Scale up next Web3D or metavers
- Accept any kind of 3D representations
 - For UGC and realistic contents
- Adaptive streaming of compressed 3D models
 - To reduce the required upload bandwidth and download delays
- VE integrated to Web, Web integrated to VE
 - Web 2.0 and Web 3D are complementary
- Metaverse should not belong to any organizations and users
 - GNU/GPL v2+

The Solipsis Vision

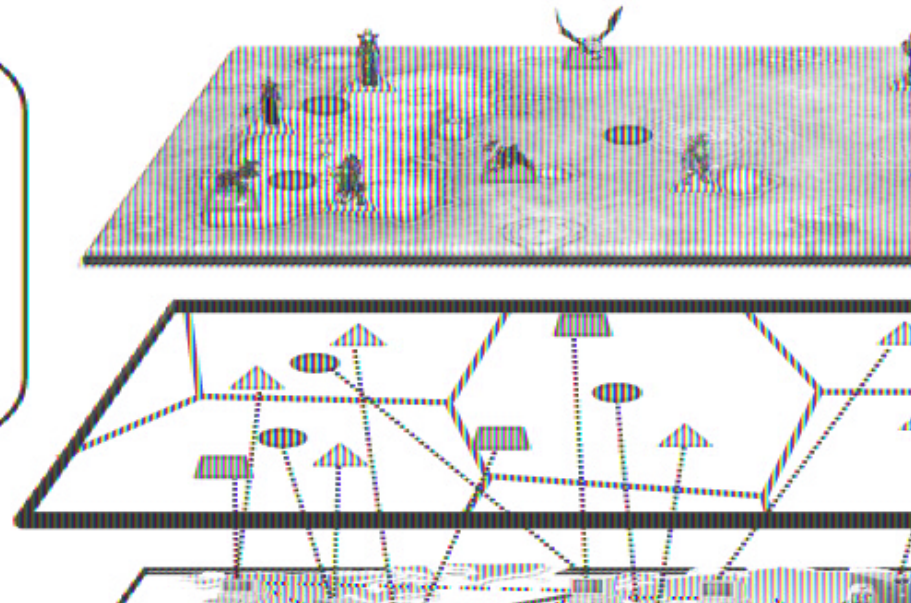
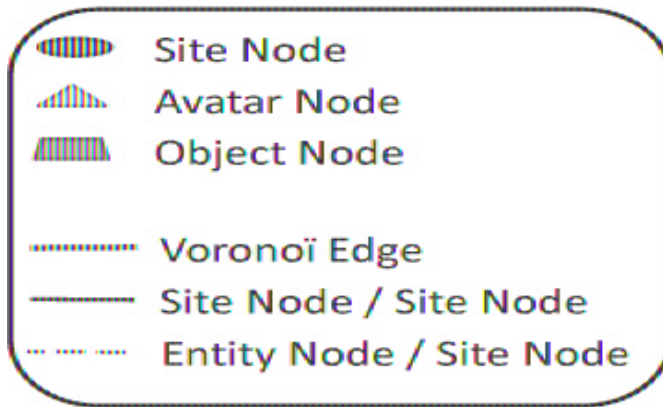
- Metaverse = set of entities
 - **Avatars** (representing the user, attached to a viewpoint)
 - **Objects** (can be moveable, interactive, picked up by an avatar, etc)
 - **Sites** (represent portion of the Metaverse occupied by avatars and objects)
- Each entity associated with
 - content (3D, video, text, etc)
 - descriptors (representing its state)
- Peer-to-Peer Architecture disseminating content and descriptors
 - Enabling scalable interaction among entities
 - Able to adapt to dynamics of large-scale environments

P2P Architecture

- Dynamically divide virtual space among peers
 - Multi-dimensional space overlay
 - Based on approximation of Voronoï tessellation
- Structure of Overlay Reflects Structure of World
 - One site < --- > One peer
 - Avatars roam around sites by connecting to corresponding peers
- Physics computation carried out only by interested parties

P2P Architecture

- RayNet Overlay



Solipsis Overlay: RayNet's Key Features

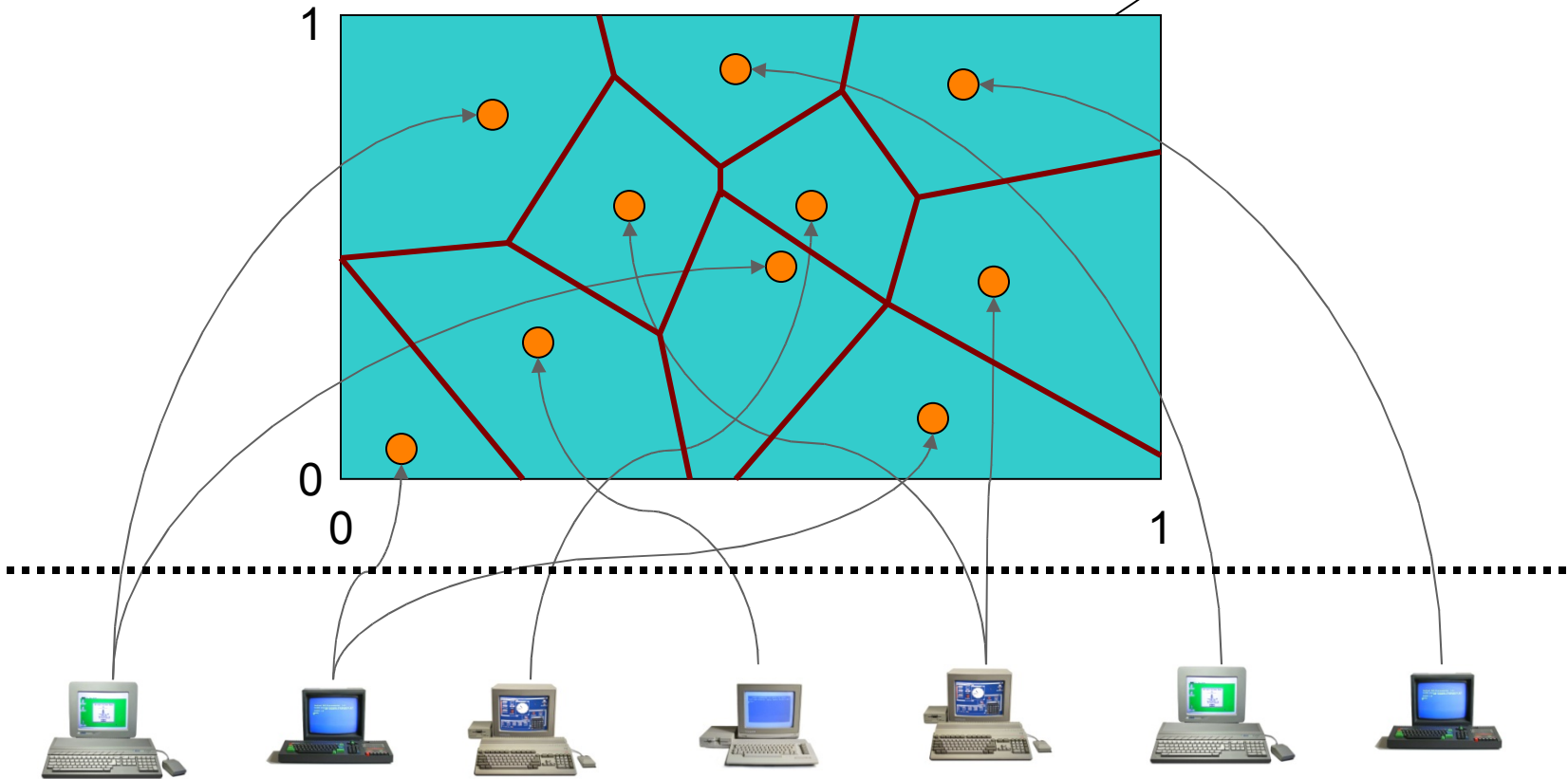


- Organize Site Nodes into Overlay Network
 - Enable fast access to peer responsible for given position
 - Ease of interaction between neighboring sites
 - Ability to adapt to disconnection of peers and a appearance of new ones
- Voronoy Tessellation of virtual space
 - Assign a tile to each peer
 - Dynamically compute neighboring relationship

● Application objects
Peers in the overlay

— Voronoï Tessellation
of the set of objects

[0:1]x[0:1] objects space



Computing entities (physical nodes)

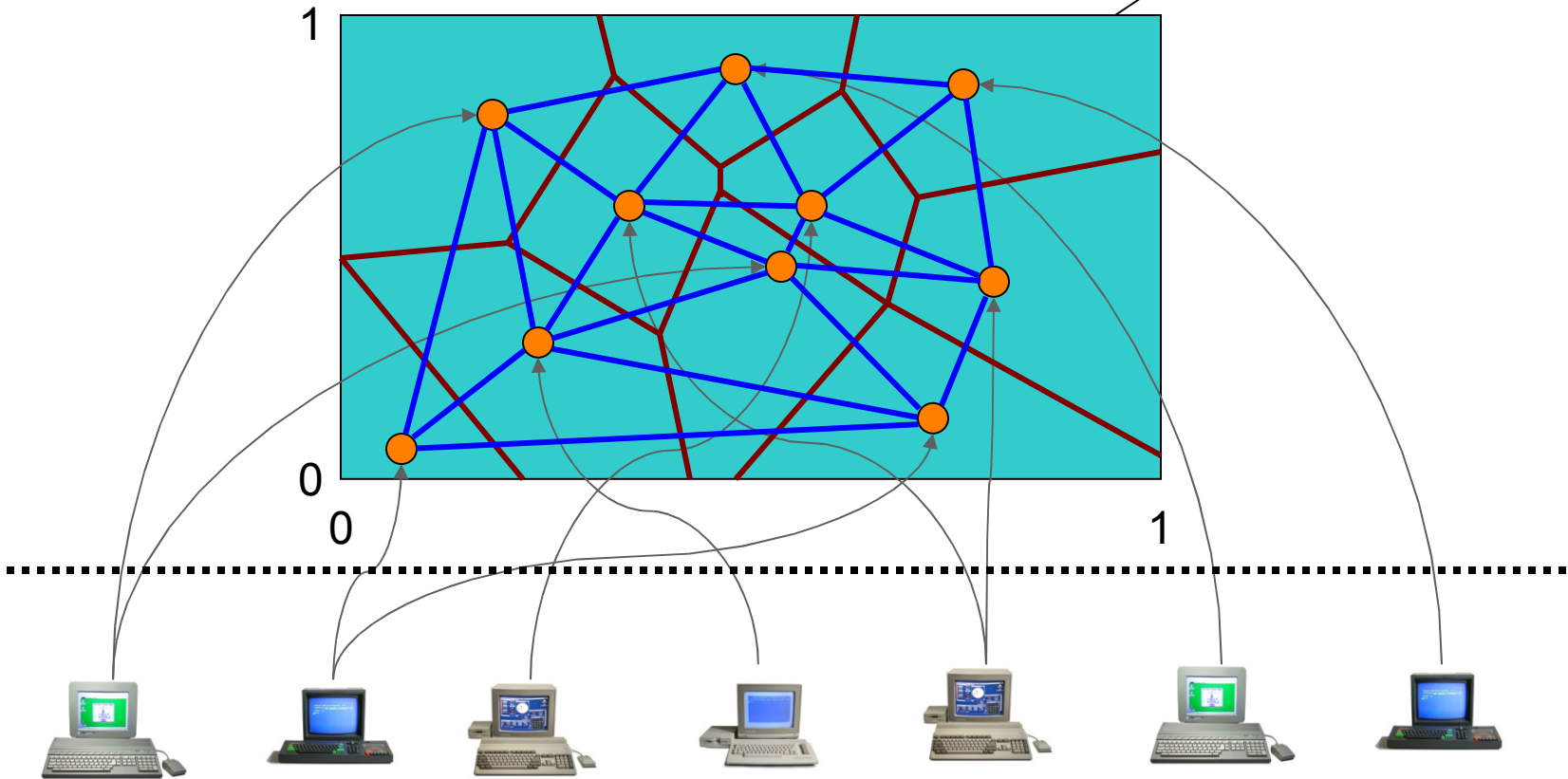
Node n_i owns o_i objects $\Rightarrow n_i$ participates o_i times in the overlay

● Application objects
Peers in the overlay

— Links between objects
(Adjacencies in the
Voronoi tessellation)

— Voronoï Tessellation
of the set of objects

[0:1]x[0:1] objects space



Computing entities (physical nodes)

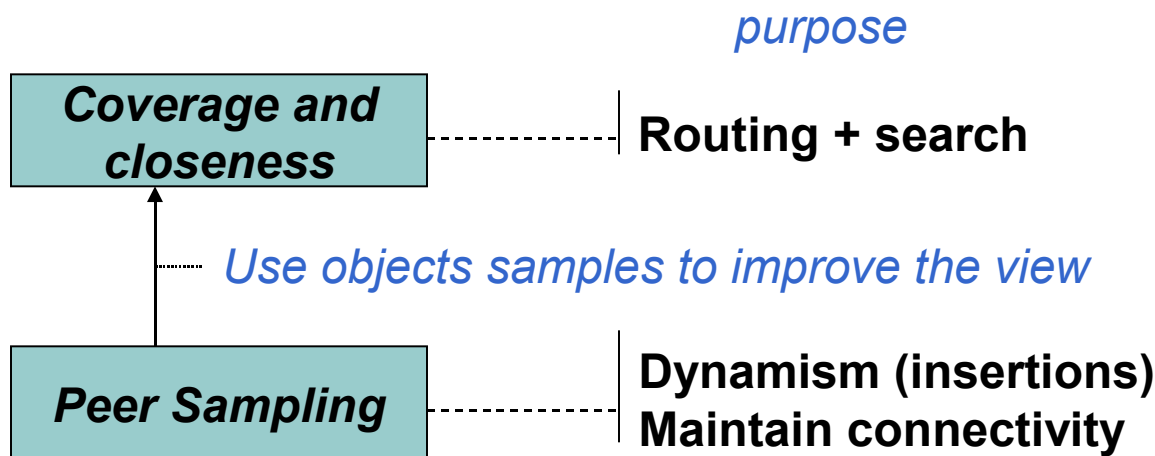
Node n_i possesses o_j objects $\Rightarrow n_i$ participates o_j times in the overlay

Why Voronoi?

- Motivation
 - Enable Greedy Routing in multiple dimensions
 - Support Expressive Search Mechanisms
- Challenges
 - Computationally Expensive
 - Maintenance in Face of Churn
- Solution
 - RayNet [Riviere 2007]

RayNet in a Nutshell

- Peer sampling
 - Gossip-based shuffling of views
- Coverage and closeness
 - Gossip-based construction of approximate Voronoï links
 - Close objects (in the “virtual” space) in all directions



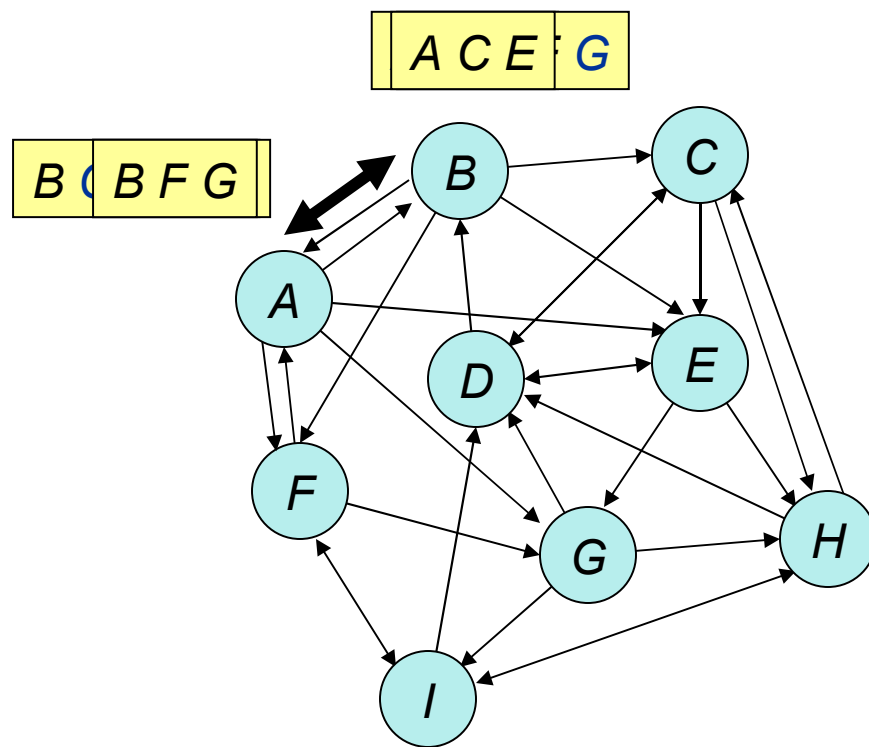
Challenge:

Evolution of local views towards a global routing structure

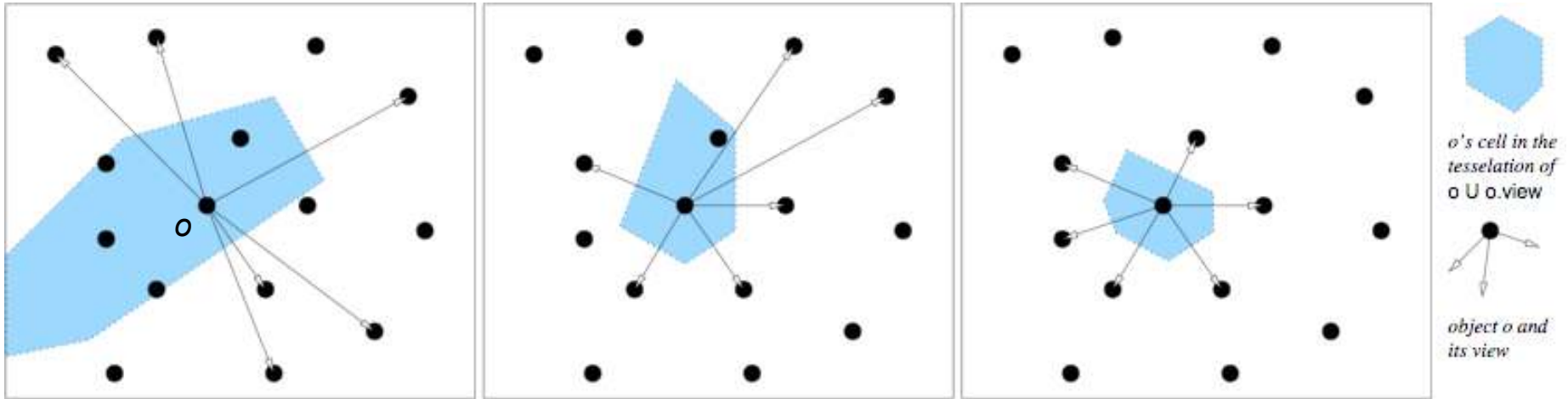
Gossip-based overlay construction



- View = set of neighbours objects
 - Fixed size c (small)
- Periodic exchange of views
 - Partner selection: LRU
 - Push/pull exchange
 - Initiator pushes itself
 - Keep c peers according to a selection function
 - Local convergence of the view's global properties



Coverage and closeness



An object o knows only its view

- o 's view == Voronoï neighbours
 \Rightarrow new objects will not change this view
 \Rightarrow the volume of o 's cell is minimal

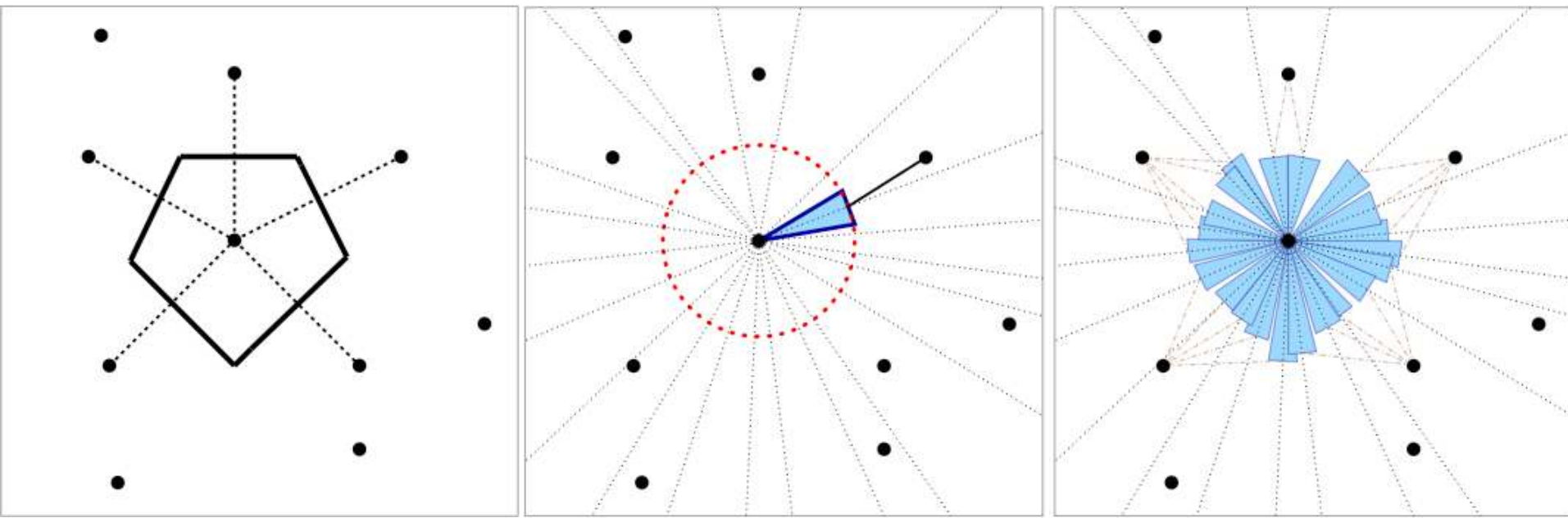
Idea: Exchange views & converge towards an approximation of Voronoï neighbours

No need to compute the Voronoï cells ! Use the **volume** as an indication of convergence

Monte Carlo cell size estimation



- Comprehensive cell topology: $O(c^d)$; volume estimation: $O(d)$
- **Idea:** sample the boundaries of the zone using “rays”
 - Each ray intersects a zone boundary
(There is a point that is equidistant to o and a neighbour)
- 1 ray = 1 ball ; $\text{mean}(\text{balls' volumes}) \approx \text{Voronoi cell volume}$



View update operation: naïve approach



- View size is $c=3d+1$ peers
- Exchange entire views : $o.view + o_{\text{partner}}.view$
- For each set S of objects of size c , in $o.view + o_{\text{partner}}.view$
 - Estimate the volume of o 's cell in the diagram of S
- Keep the set with minimal volume as the new view
- Effective, but there are $O(c!)$ configurations to examine...

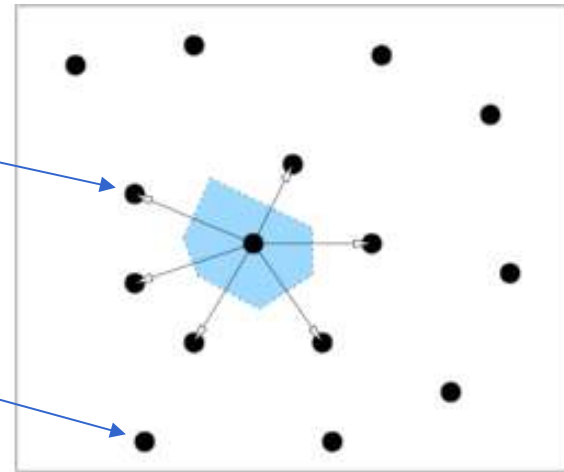
View update operation: efficient approach



- Determine the potential contribution of each object to the coverage and closeness (ie to the volume of o's cell)
- For each object o' in o.view + o_{partner}.view
 - Calculate the volume of o's cell in o.view + o_{partner}.view without o'

*Ignoring this object results
in a bigger zone:
High contribution*

*Ignoring this object does not
impact the size of the zone:
No contribution*



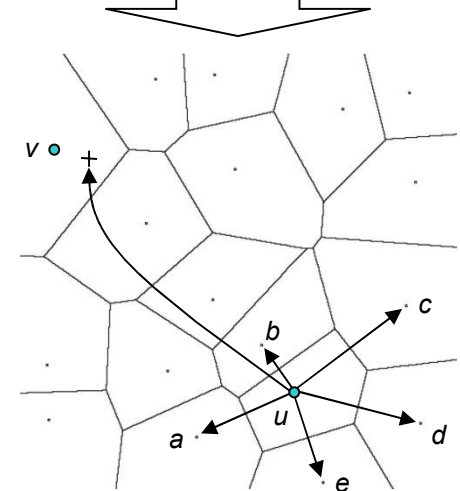
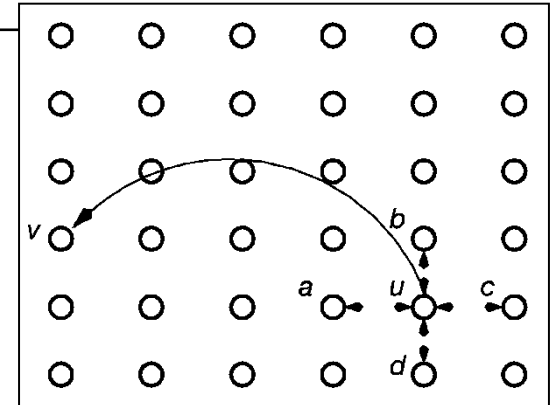
- Keep the c objects with the greatest contribution

Enabling efficient routing

- Routing in the approximate Voronoï diagram requires $O(N)$ hops
- Small-Worlds models:
 - Small paths + navigability
- Kleinberg model
 - Grid + long links with d-harmonic distribution of lengths
- Using biased peer sampling:

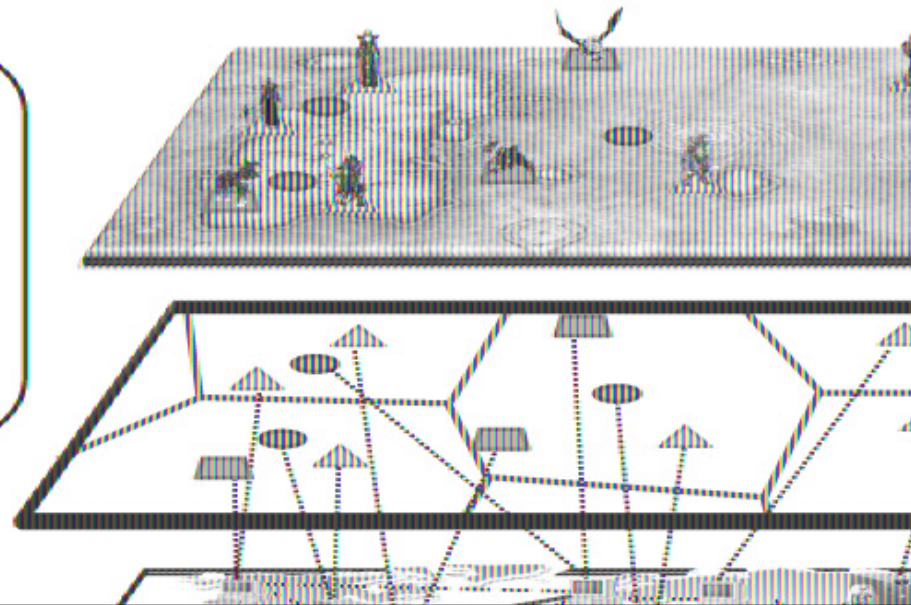
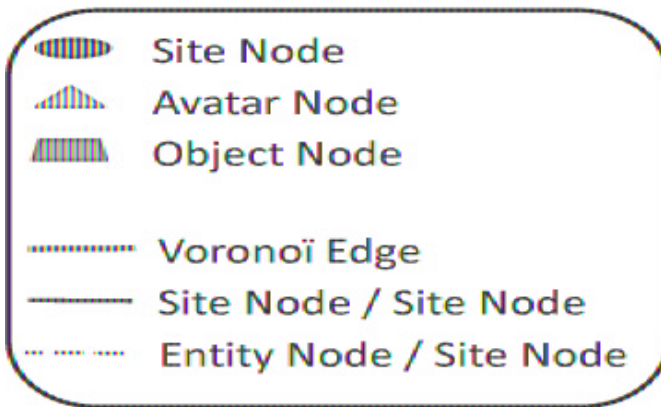
(by Bonnet/Kermarrec/Raynal, OPODIS07)

 - Similar long link length distribution
- $O(\log^d N)$ routing with 1 long link



Back To Solipsis

- Exploiting the RayNet Overlay



Exploiting the RayNet Overlay



- Each peer may manage sites and avatars
 - Organize Sites into Voronoi Structure
 - Avatars communicate with sites in which they roam
- Adaptive Update Exchange according to
 - entity type: reduce site communication
 - entity proximity: ensure interactivity between close entities

Adaptive Update Exchange

- Between **Sites**
 - Low frequency with multi-hop propagation
- Between **Site** and **Avatar/Object** entities
 - Low frequency
 - All Avatar/Object Entities into a Site's cell communicate with it
 - Entities have a knowledge of their neighborhood
- Between close **Avatar/Object** entities
 - High frequency
 - heartbeat mode (propagation)
- Between very close **Avatar/Object** entities
 - Very high frequency
 - For Collision detection

Decentralized physics computation

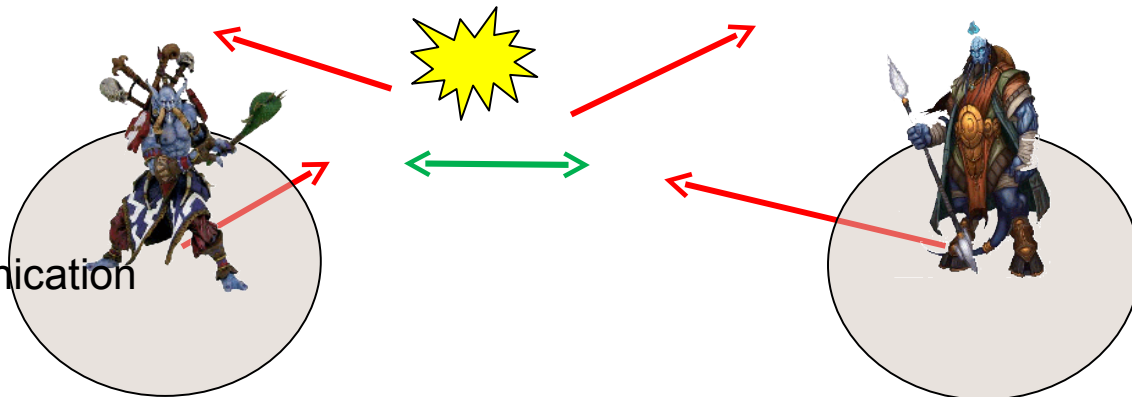


- Each entity owner computes its own position according to physical properties
 - Mass, momentum
 - Forces applied by entities in its surrounding
- Collision Detection Area (CDA) defined by a sphere
- Entities having their CDA in collision launch CD communication

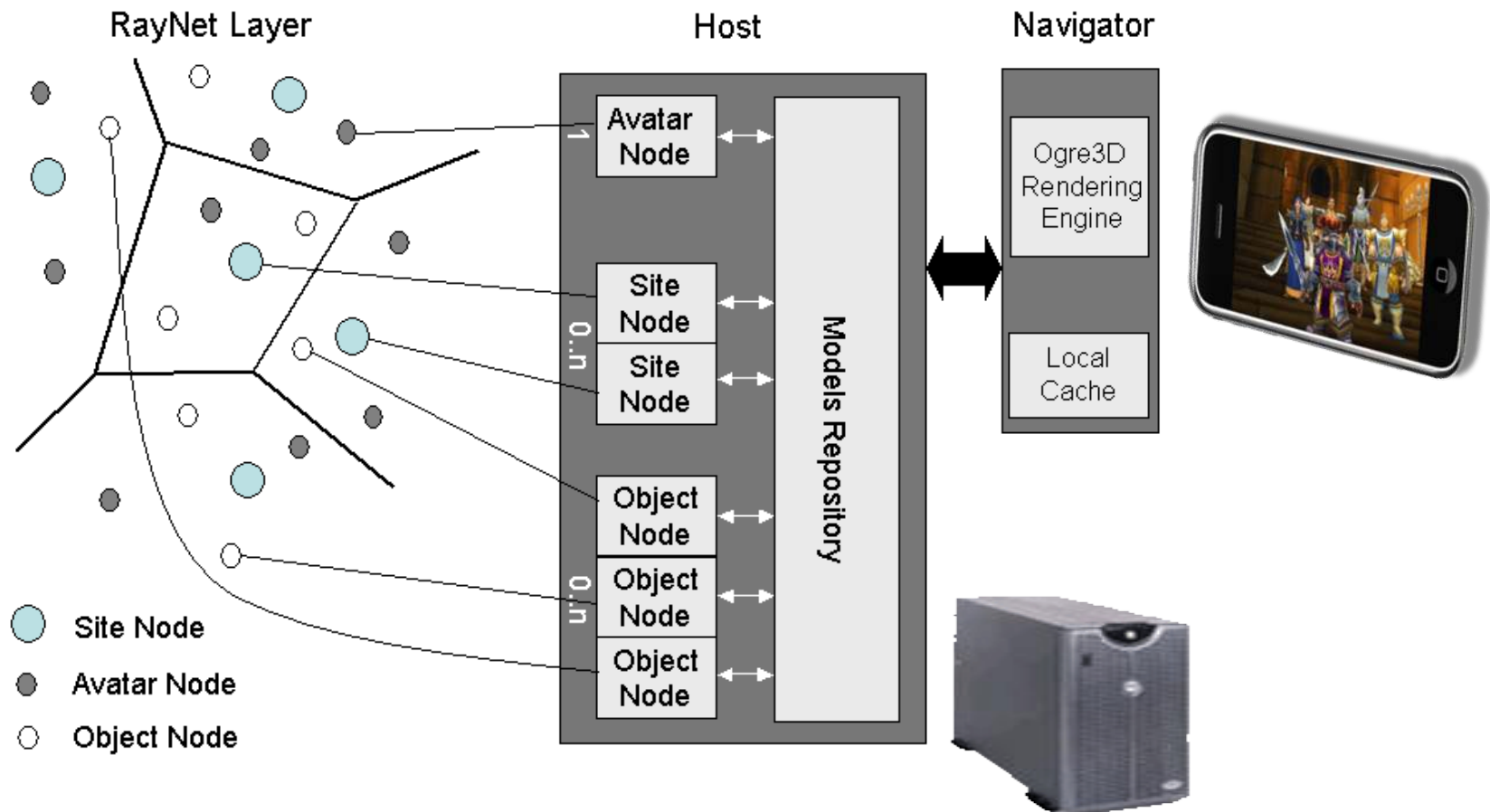
→ Force

○ CDA

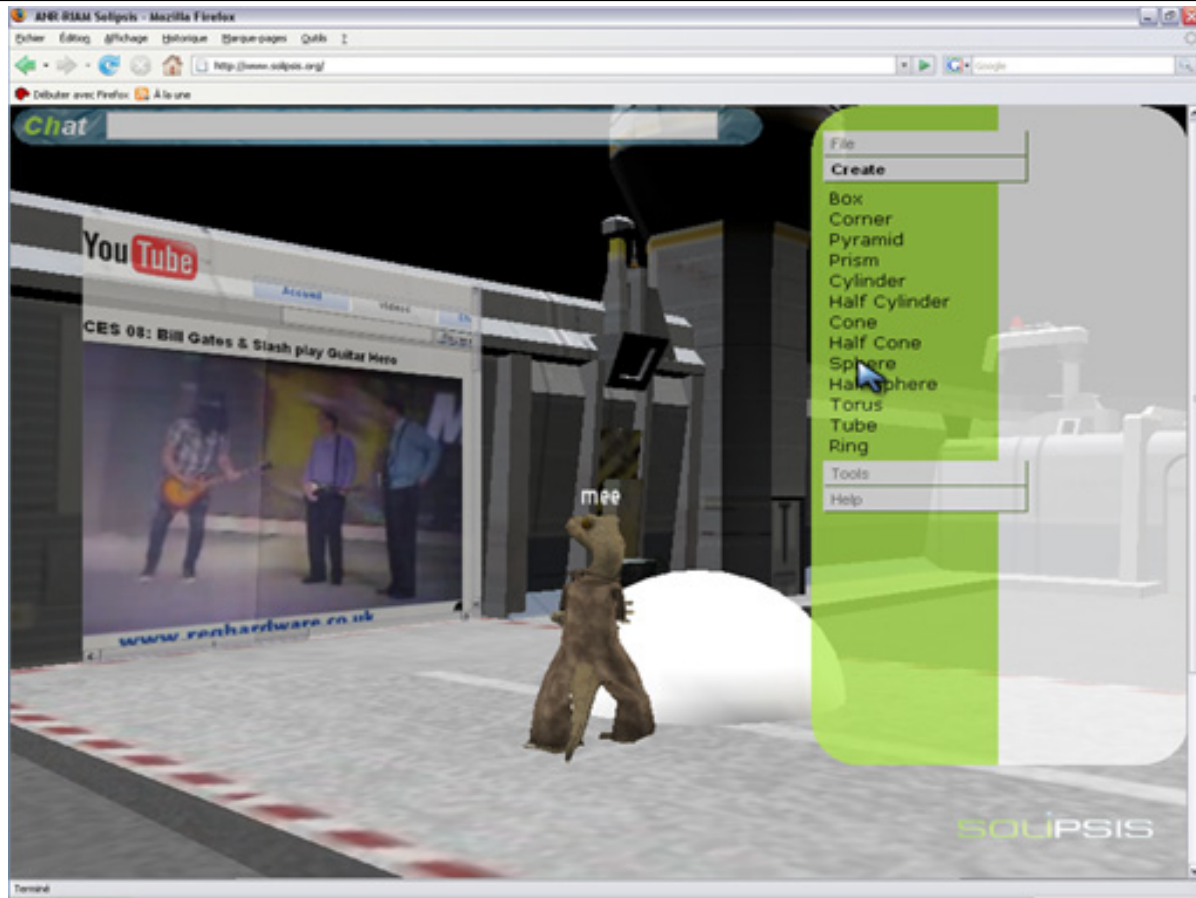
↔ Collision Communication



Navigating Solipsis



Navigator



Current Status And Vision

- Ongoing Open-Source Implementation of Solipsis
 - Navigator
 - RayNet Layer
 - Physics Computation and Communication Protocol
- Tackle new issues
 - Persistency Management
 - Privacy and Malicious Nodes
- More Information on www.solipsis.org



Thank You

Descriptors

- Example of a simple descriptor

UID	universal identifier of the entity
seqNum	sequence number
owner	identifier of the node managing the entity
type	site, avatar or object
loc	location in the 3D space
ori	orientation in the 3D space
shape	shape from a predefined set
box	bounding box of the object
R_p	perceptibility radius: distance from which the object is visible in the absence of obstacles
R_b	radius of smallest sphere enclosing the entity
objs _a	list of entities attached to the current one
f_1	first file of 3d-description
v_1	version number for first file
c_1	list of hosts that have cached v_1 of f_1
...	...
f_n	n-th file of 3d-description
v_n	version number for n-th file
c_n	list of hosts that have cached v_n of f_n
...	additional fields for progressive levels of details

Dynamic Object Management



- Animated Objects have an animation script
- Every hosts (PC, server, etc) can take the responsibility of an object (according to its rights), and animate it
- If an avatar uses an object, its host is responsible of the object's animation
- If avatar throws an object (with its animation script), object becomes passive and animated by site node