Hibernation of Advanced Railroad Trains (ArrT)

Step Three – … and Communicate Them

New Concept Paper of SrrTrains v0.01 **- UNFINISHED**

The „Hibernation of ArrT“ consists of three Steps.

During these steps the project SrrTrains v0.01 will not do any external progress, just internal improvements will be done in an invisible manner:

1. **Step One** …... Relax and Clean Up,
2. **Step Two** …... Develop Visions...
3. **Step Three** …... ...and Communicate Them.

Currently we are developing **Step One „Relax and Clean Up“** (which will be accompanied by Release 0033.10 „Pieta“ of the software, as soon as finished).

The present hibernation report, which is the „New Concept Paper of SrrTrains v0.01“ will be finished as a sub-step of Step Three, more or less together with LAN Party #3.

Table of Content

1 A Vision of the 3D Web – How To Enter the Internet

The present chapter will try to elaborate the vision behind SrrTrains

Tbd.

2 Composite Scenes – the Idea DIGITS

The present chapter will try to describe the ideas that I was collecting between the years 2000 and 2005

Tbd.

3 Multiuser Scenes – the Idea SIMUL-RR

The present chapter will try to describe the idea that drove me towards multiuser scenes

Tbd.

4 The Network Sensor

The present chapter will try to describe the basic ideas of the Network Sensor / Event Stream Sensor, as available at the Web3D Consortium

Tbd.

5 SrrTrains Community, SrrTrains Core Team, Me

The present chapter will try to describe the setup of the SrrTrains v0.01 project

Tbd.

6 Client Based Server Software / The BIMPF Approach

The present chapter will try to describe the characterisitics of how to use the Network Sensor / Event Stream Sensor within SrrTrains

Tbd.

7 The Simple Scene Controller

> The present chapter will try to describe the first X3D prototype that was a part of the SMUOS Framework

Tbd.

8 The MMF Paradigm

> The present chapter will try to explain, why the SMUOS Framework requires to decompose the scenes into models, modules and frames

Tbd.

9 MAM, OBCO and MIDAS Objects

> The present chapter will try to explain some very basic concepts of the SMUOS Framework

Tbd.

10 Unbound Models and Handover

> The present chapter will try to explain some additional concepts of the SMUOS Framework

Tbd.

11 The MIDAS Base

> The present chapter will try to explain the properties that are common to all MIDAS Objects

Tbd.

12 Moving Modules – the eMMF Paradigm

> The present chapter will try to explain some future concepts of the SMUOS Framework. Those concepts will not be implemented, even if the New Concept Paper will be finished

Tbd.

13 3D Graphics and the Theory of Relativity

The present chapter will try to explain some future concepts of the SMUOS Framework. Those concepts will not be implemented, even if the New Concept Paper will be finished

Tbd.

Appendix A The SMS Tracer

The present chapter will try to explain the SMS Tracer

Tbd.

Appendix B The Console Interface

The present chapter will try to explain the Console Interface

Tbd.

Appendix C Extensibility of the SMUOS Framework

The present chapter will try to explain the interface that is provided by the SMUOS Framework in order to be extended by SMUOS Extensions

Tbd.

Appendix D Message Flows – Best Current Practices

The present chapter will try to explain some concepts by examplifying them

Tbd.

Appendix E Reality – the (N+1)th Scene Instance

> Anything is real.
> There is nothing that does not exist.
> See also the glossary (Appendix G Glossary).

Inspired by a posting on X3D-public, I decided to think about an additional use case of the SMUOS Framework in February 2011. This thinking has not yet been closed.

Given, a multiuser session consists of N scene instances (N >= 1), what about adding an (N+1)th scene instance that would not be inhabited by users, but that would connect the multiuser session to the reality.

Hence we would have N **virtual** realities – one for each user – and one **real** reality, all of which would be synchronized to each other.

Thus we would have following operational modes for the multiuser session:

- Single user mode (N = 1, SCSI does not exist)
- Multi user mode (N > 1, SCSI does not exist)
- Mixed reality mode (N >= 1, SCSI exists)

The present chapter tries to examplify all of those operational modes.

The overall picture is as follows:

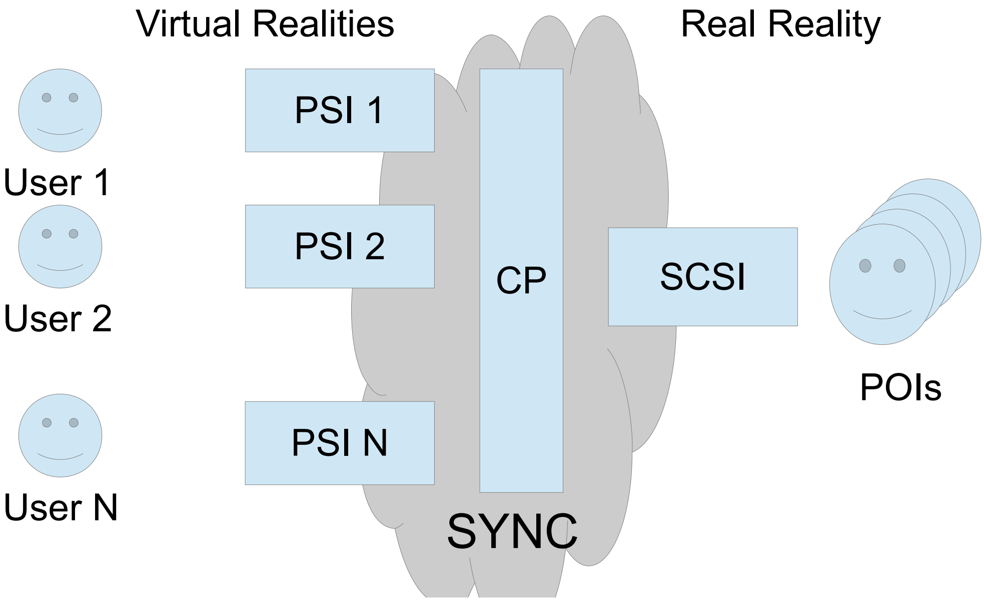

Figure 1: Overall picture of a multiuser session

E.1 Example of a Singleuser Session

An instance of a Simple Multiuser Scene (SMS) can run in single user mode.

Figure 2 depicts all interactions that are possible in single user mode.

Example:

A **user** (e.g. „Alice“)

employs her **senses and skills** (e.g. the abilities to move the mouse and to press buttons on the keyboard or to view the screen) to exchange information with a **personal scene instance** (PSI) – i.e. to inhabit the SMS.

Hence the user attains a **virtual identity** (e.g. „Gandalf“)

and the ability to employ **virtual senses and skills** (vSaSk), e.g.:

- to inhabit the SMS,
- to touch a model,
- to take the ring,
- and so on.

The PSI renders avatars, models and modules, to represent the facilities of the SMS, i.e.

- an **avatar** to represent the virtual identity, e.g. „Gandalfs Avatar“,
  to be exact, we can call it a **virtual life avatar** (VLA),
- **models** to represent the renderable objects of the SMS, e.g. „the ring“,
- **modules** to represent the surroundings of the SMS, e.g. „Mordor“.

Note:

The avatar does not represent the user, but it represents the virtual identity. A small but important difference.

An SMS might enable a user to change the virtual identity without logging out, or it might even enable the user to attain more than one virtual identity in parallel.

Note:

In our example, the mouse, the keyboard, the screen and so on are parts of the personal scene instance (PSI). The interface of the PSI can be used directly by the senses and skills of the user.

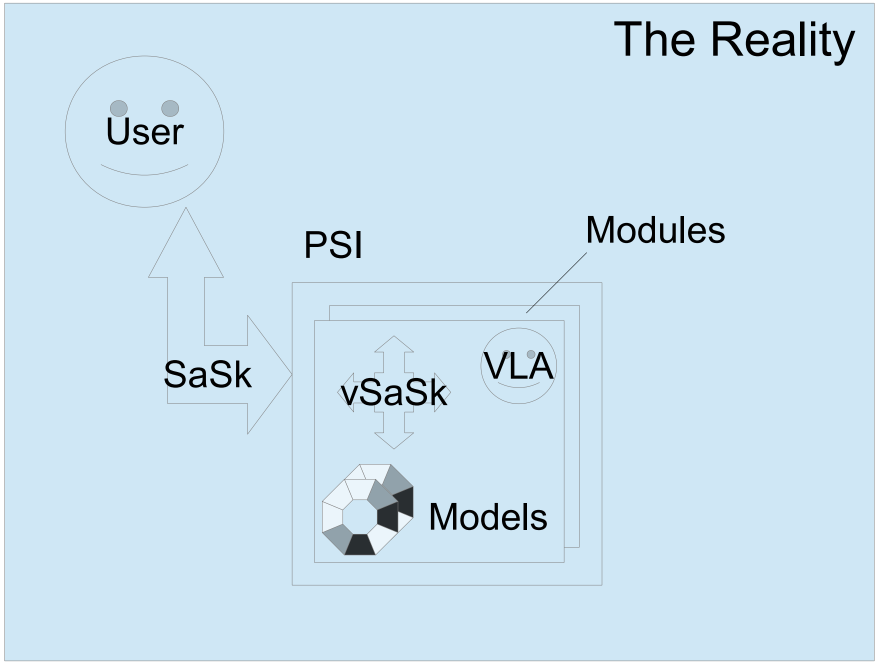

Figure 2: Possible interactions of a multiuser session in single user mode

E.2 Example of a Multiuser Session

A set of instances of an SMS can run in multi user mode.

Figure 3 depicts all interactions that are possible in multi user mode (shown for 2 users).

Running in multi user mode means, every instance of the SMS that runs within the multiuser session, exchanges information with each other instance that runs in the same multiuser session, hence synchronizing the state of all modules, models and avatars and hence keeping the same virtual reality for all N users of the multiuser session.

Example:

Synchronizing the State can mean synchronizing e.g.

- position
- orientation
- door open / door closed
- light on /light off
- and so on

Note:

Users may join or leave the multiuser session arbitrarily, hence N is not constant over time.

Note:

In addition to inhabiting the virtual reality, it may be relevant that all users inhabit the reality, too.

E.g., some of the users could be logged in to a chat room and chat with each other or they could be members of a phone conference to exchange information and/or emotion via voice.

As long as those facilities – we call them **real life facilities** (RLF) – are separate from the facilities of the SMS, we call the multiuser session an **application of virtual reality**.

Otherwise, if real life facilities are synchronized with facilities of the SMS, then we would call the multiuser session an application of **augmented** reality, of augmented **virtuality** or generally spoken of **mixed** reality, see next section.

Note:

Outside of the virtual reality, **users** represent their **real identities**, within the virtual reality, **avatars** represent the **virtual identities**.

We will see an example in the next section, where a **real life avatar** (RLA) represents a **virtual identity** within the **real reality** – this is really cool stuff :-).

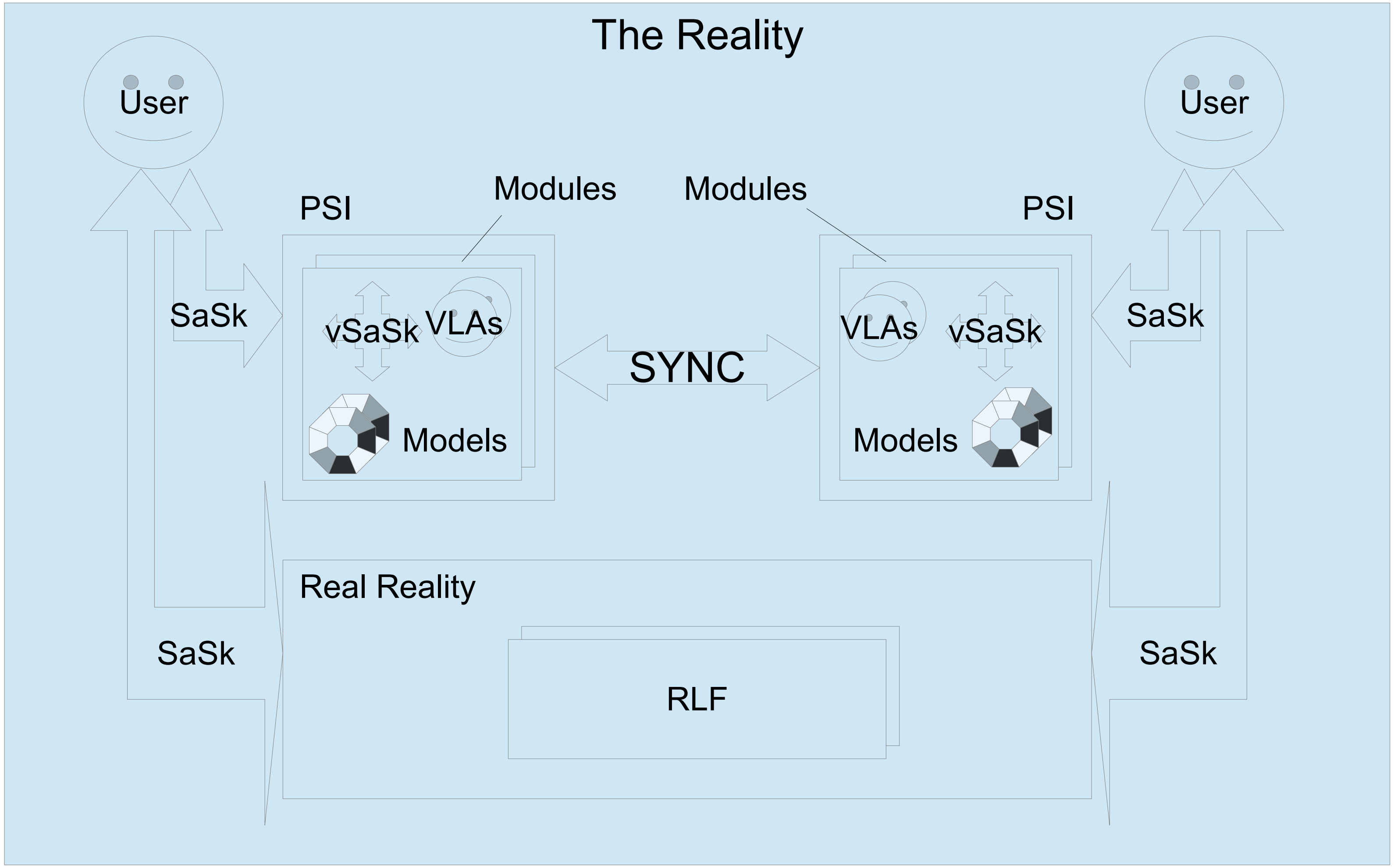

Figure 3: Possible interactions of a multiuser session in multiuser mode (2 users shown)

E.3 Example of a Mixed Reality Session

A set of instances of an SMS can run in mixed reality mode, if at least one facility of the SMS represents a real life facility.

Figure 4 depicts all interactions that are possible in mixed reality mode (shown for 2 users).

Example:

> In mixed reality mode we do not use the term real life facility (RLF) – which would be too general – but we distinguish three kinds of RLFs:
>
> - **Real life avatars** (RLA), e.g. a drone,
> - **real life objects** (RLO), e.g. airplanes that do not represent virtual identities, and
> - **collateral entities** (CE), e.g. a stealth drone of the enemy, as long as not detected.
>
> Now the users attain **remote senses and skills (rSaSk)** that are provided by the RLAs (and optionally by any RLOs) e.g.
>
> - A drone (an RLA) provides the rSaSk of destroying real life facilities
> - A remotely controlled car that is modelled in the SMS but that does not represent a virtual identity (i.e. an RLO) provides the rSaSk of crashing into a house
>
> RLFs are **synchronized** with VLFs:
>
> - The state of RLAs is synchronized with their VLAs, e.g. position and orientation
> - The state of RLOs is synchronized with their models, e.g. position and orientation
> - The state of modules is derived from the universal positioning system (UPS).
>   If we assume, e.g., a module being the inner part of a railway wagon, then the wagon spans a local coordinate system that imposes an acceleration to all of its inhabitants (which is equivalent to a gravitational field). Hence the properties of the module are synchronized with the properties of the real wagon (an RLO). The module is a part of the model and the RLO has got UPS properties, it contains a level of the UPS.

Note:

> Real life objects (RLOs) and real life avatars (RLAs) are modelled in the SMS and their states are synchronized with the states of the models and virtual life avatars (VLAs).
>
> Collateral entities (CEs) are NOT modelled in the SMS.

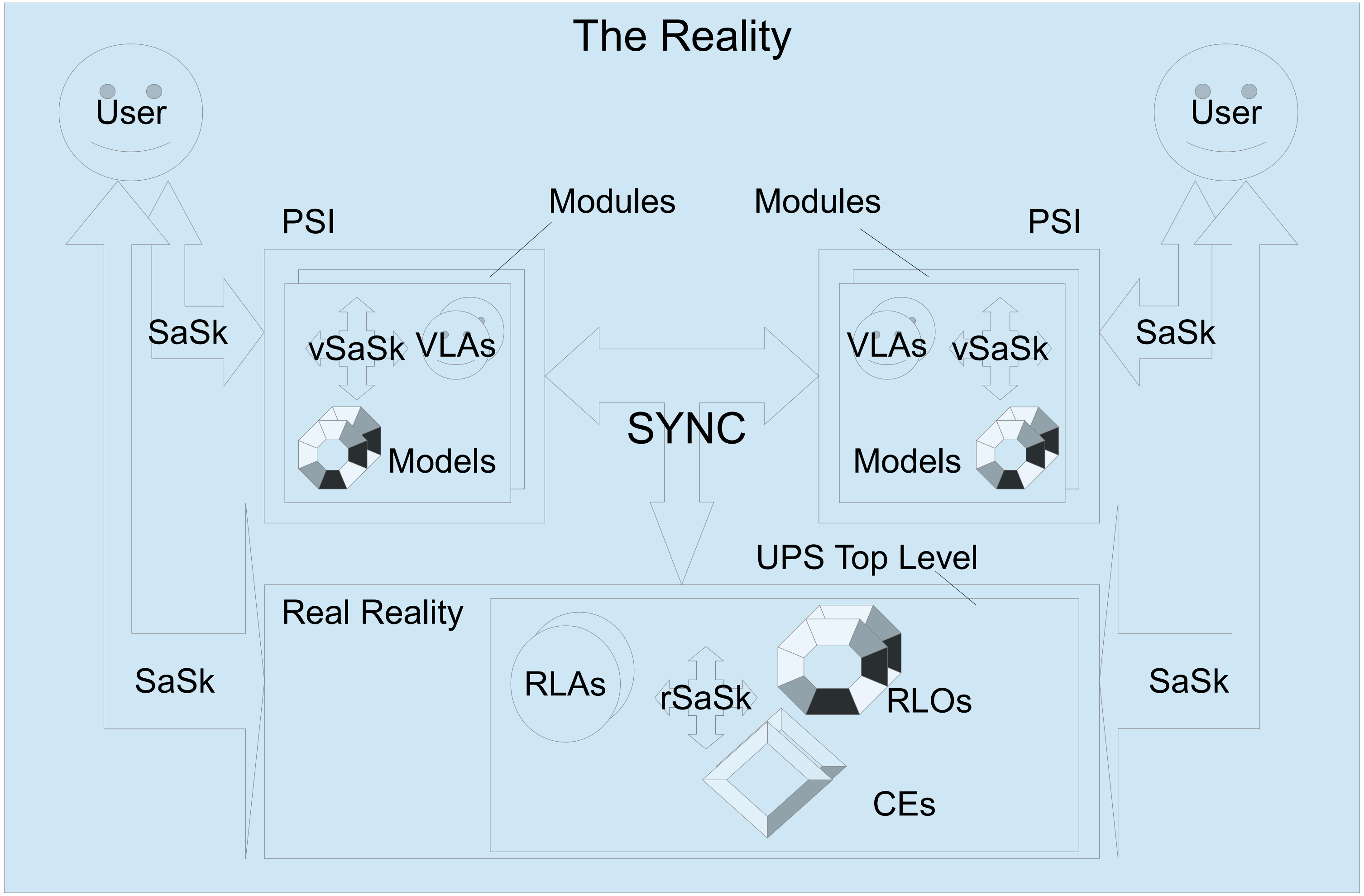

Figure 4: Possible interactions of a multiuser session in mixed reality mode (2 users shown)

E.4 A few Words about Synchronization

In the latest examples we stated the PSIs of a multiuser session are synchronized to each other, if the multiuser session runs in multiuser mode.

In mixed reality mode we would just have an additional scene instance, which would not be a PSI, but which would be the SCSI.

Nice words, sounds simple.

However, synchronization is a little bit tricky, if we want to understand the whole story.

First, we have to understand the **events** and **states:**

An **event** is singular in that it does not depend on history.

Something happens in one scene instance and the scene instance informs all scene instances that this thing has happened and that's it.

Example:

> User „Alice“ touches a model and the model starts an animation in all scene instances (that of „Alice“ and that of „Bob“).

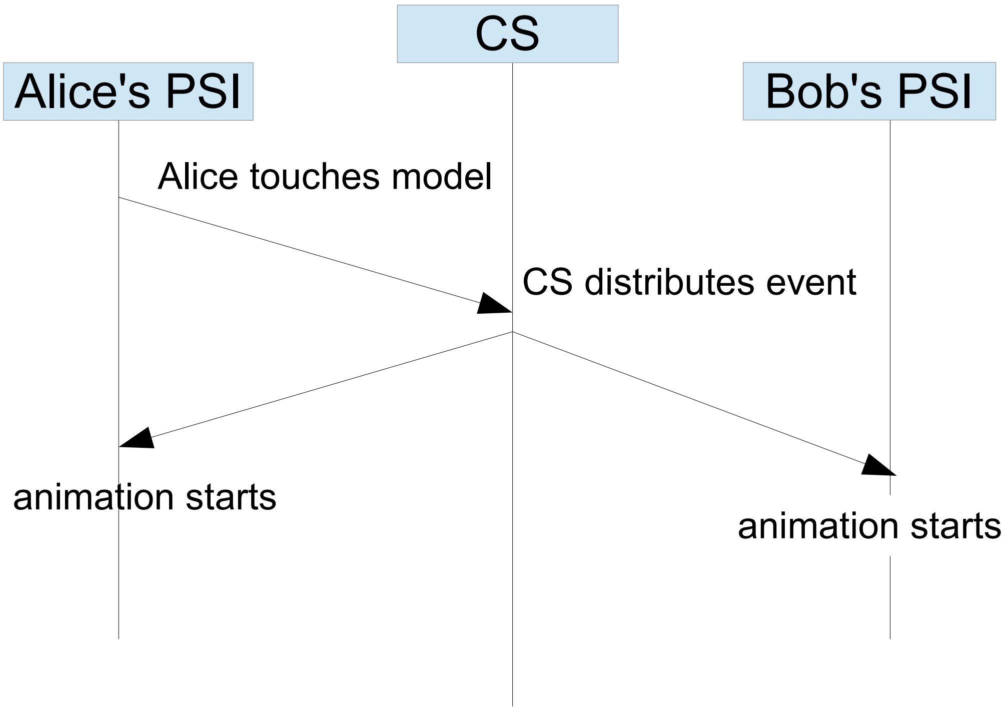

Figure 5: Example of the distribution of an event

A **state** is stored persistently on the collaboration server (CS).

Changes of a state are relative to the current value of the state.

Changes are implemented by **server side calculations**.

Example:

> „Alice“ resets the position of a car.
>
> Then „Bob“ joins the session, gets the current value of the state during initialization and afterwards changes the position of the car again.

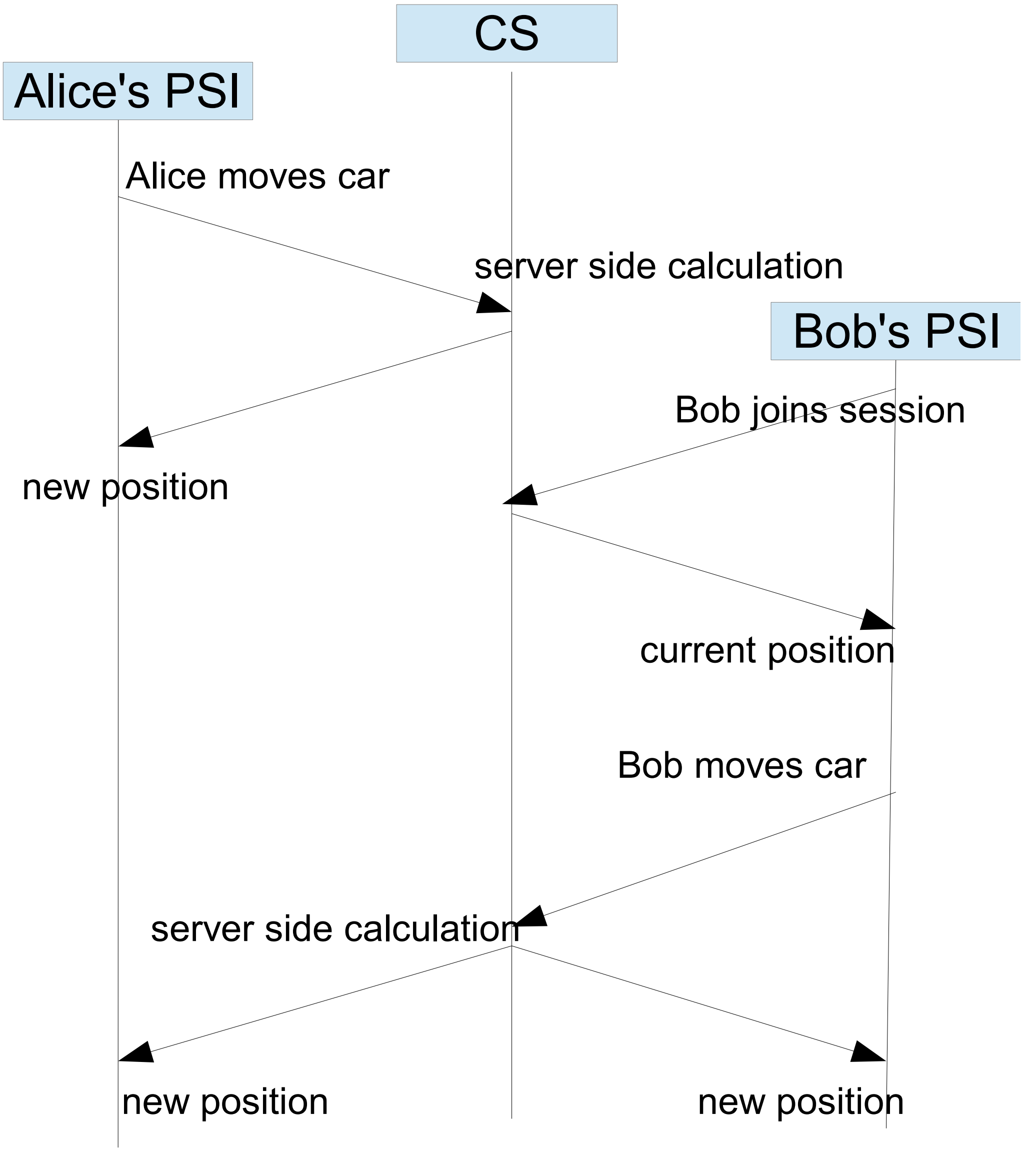

Figure 6: Example of changing a state with server side calculations

Second, we have to understand **MIDAS Objects:**

MIDAS Objects add one level of indirection to the principles of states and events.

That means, the calculations are not performed as server side calculations but they are performed as client side calculations.

The CS is only used to cache the global state of a MIDAS Object. The state is actually stored in the scene instances in the instances of the MIDAS Object.

Only one scene instance is responsible to calculate the global state of a MIDAS Object. We say this instance has got the **Object Controller Role** (OBCO Role), or shortly **it is the OBCO for this MIDAS Object.**

Example:

„Alice“ changes the position of the car. „Bob“'s PSI is the OBCO of the car.

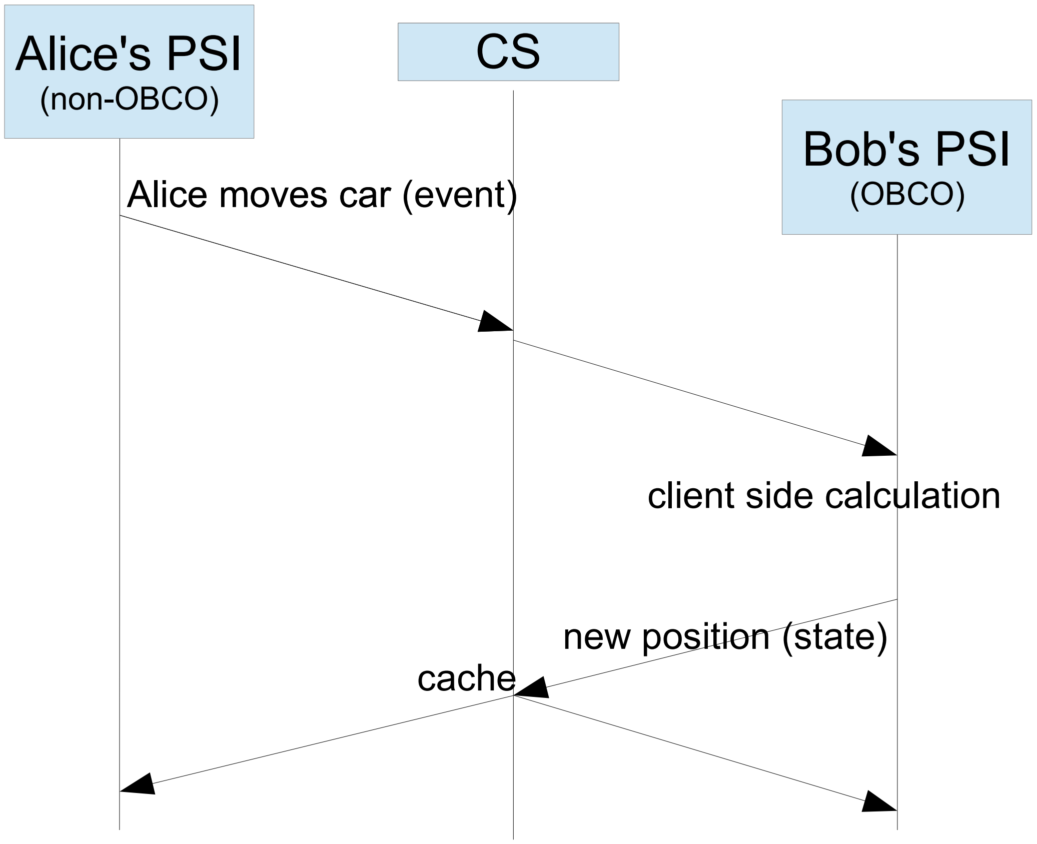

Figure 7: Example of changing a state with client side calculations

We can depict the principle of MIDAS Objects in a layered diagram

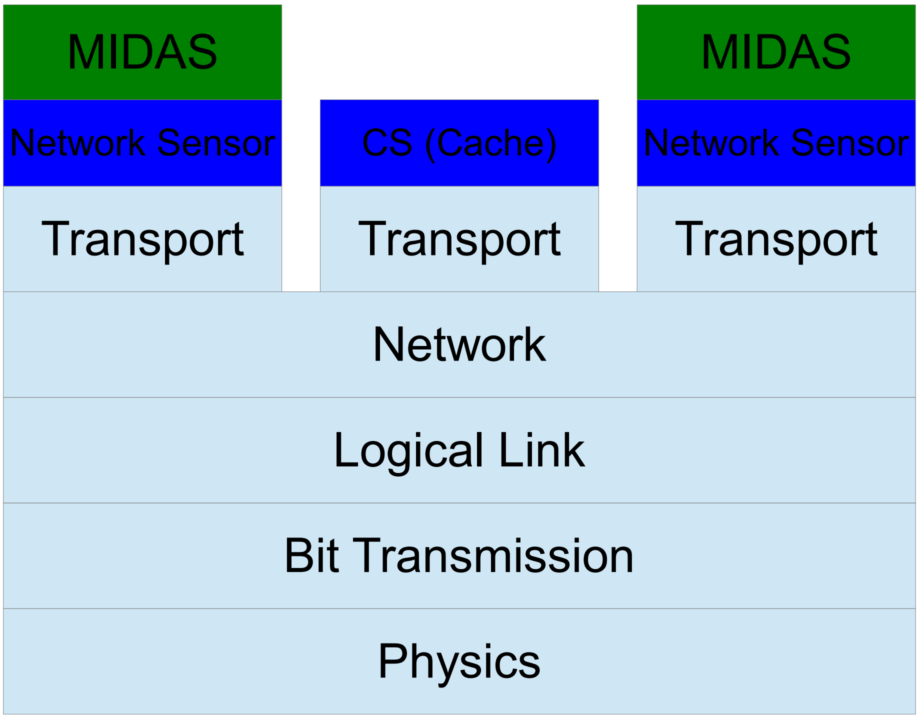

Figure 8: Layered view of MIDAS object and CS within multiuser session

Third, we have to understand **POIs**

POIs are the real life counterparts of MIDAS Objects. MIDAS Objects reside in the 3D Web, POIs reside in the IoT. This is ffs.

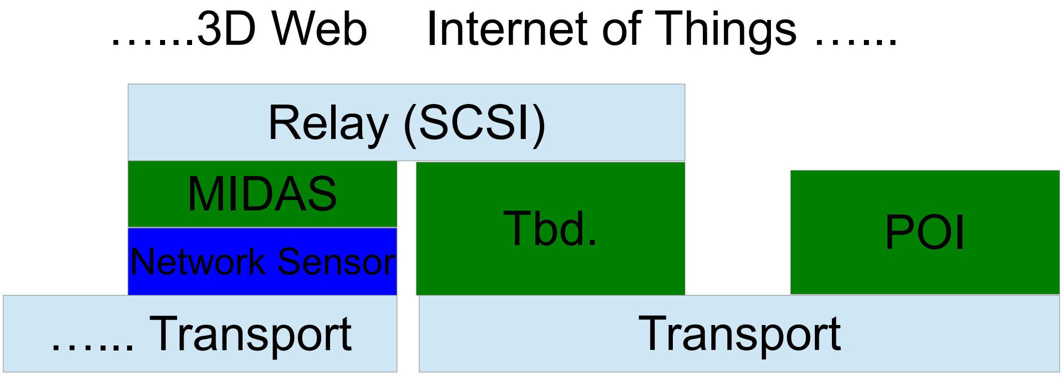

Figure 9: The relation between MIDAS Objects and POIs

Appendix F Trying a Prophecy about SrrTrains v0.01

Tbd.

Appendix G Glossary

Reality, Virtual Reality, Real Reality

There is only one **reality**, but every person carries an own **model of the reality** in his mind. This model helps the person to foresee the future development of reality and it helps the person to influence reality according to his will.

Virtual reality is a part of the reality that is implemented by means of technology and that helps one or more persons (see **user**) to inhabit a virtual scene that needs not be directly related to the reality.

Strictly spoken, an ancient form of virtual reality is already to sit around the camp fire telling stories. Also books and movies form kinds of virtual reality.

Usually we use the **narrow term virtual reality**, if some minimum technological requirements are fulfilled, e.g. the usage of stereoscopic computer graphics.

We use the term **real reality** to denote all parts of the reality that are not part of the virtual reality in question, but that are of relevance for the virtual reality.

Anything else is just reality.

User

A **user** is a person who uses a personal scene instance (see below) to inhabit a Simple Multiuser Scene (see below) in the course of a multiuser session (see below).

Personal Scene Instance (PSI)

A personal scene instance is the collection of all technological facilities that are needed so that one user can inhabit a Simple Multiuser Scene.

One important facility of the PSI can be a Web3D browser that interprets a concrete scene graph.

The user interface of the PSI can be used via the senses and skills (SaSk) of the user.

Simple Multiuser Scene (SMS)

A Simple Multiuser Scene is a collection of facilities that are accessible via the 3D Web and that can be instantiated within PSIs to provide virtual senses and skills (vSaSk) to users.

Such facilities include, e.g. (see below for detailed definitions):

- Avatars to be able to represent virtual identies
- Models to be able to render the renderable objects of the scene
- Modules to be able to render the surroundings of the scene
- Geographic infrastructure to be able to render the surroundings of the scene

Multiuser Session

A multiuser session is an instantiation of an SMS for a concrete set of users.

Those users will be able to inhabit the virtual scene together.

Technically spoken, a multiuser session is a collection of one or more PSIs and of one optional SCSI (see below), all of which are synchronized to each other.

Server/Controller Scene Instance (SCSI)

The Server/Controller Scene Instance connects the multiuser session to the real reality in order to synchronize real life facilities (see below) with virtual life facilities (see below).

This enables the mixed reality mode to be used as operational mode (see below).

Virtual Life Facility (VLF)

Virtual life facilities are used to provide virtual senses and skills to a user. In mixed reality mode VLFs may be synchronized to real life facilities (see below).

A VLF is an instantiation of a facility of the SMS.

Examples of VLFs are:

- Virtual life avatars (or simply avatars) to represent virtual identities to one user
- Models to render the renderable objects of the scene to one user
- Modules to render the surroundings of the scene to one user
- Geographic infrastructure to render the surroundings of the scene to one user

Real Life Facility (RLF)

Real life facilities are parts of the real reality.

We distinguish following kinds of RLFs: real life avatars (see below), real life objects (see below) and collateral entities (see below).

Operational Modes (OM)

A multiuser session can operate in one of following modes:

- Single user mode – only one PSI exists, SCSI does not exist
- Multi user mode – more than one PSI exist, SCSI does not exist
- Mixed reality mode – at least one PSI exists, SCSI exists

Model, Real Life Object (RLO)

A model is an object within an SMS that can be rendered.
In other words, it is an object to the virtual senses and skills of the user, when he inhabits the SMS through the PSI.

In mixed reality mode, a model may represent a real life object (RLO).

An RLO is always represented by a model, otherwise it would be a collateral entity.

Avatar, Virtual Life Avatar (VLA), Real Life Avatar (RLA)

An avatar is an object that represents a virtual identity (see below). A virtual life avatar is a model that represents a virtual identity and a real life avatar is an RLO that represents a virtual identity.

Collateral Entity (CE)

A collateral entity is an RLF that is not an RLO. I.e. it is a real life facility that is somehow relevant for the multiuser session, but it is not modelled in the SMS.

Module, Universal Positioning System (UPS)

According to the MMF paradigm, an SMS consists of one or more modules that build the surroundings of the scene, whereas each renderable object (each model) is assigned to one of the modules.

A module spans a local (pseudo-) euclidean spacetime, which is used to position the models.

In mixed reality mode, we will often use WGS84 coordinates as global coordinates, which can be used to position the modules.

Hence a local coordinate system in real reality can be defined relative to GPS coordinates.

Now the SMUOS framework aims to be a framework for the 21st century and hence a GPS will not be enough. We will need something that includes the universe into its concepts, not only the globe.

UPS the right wording for such idea.

And it need to be hierarchical, according to the eMMF paradigm. One level of modules being the top level (within a scene) containing top level models. Each top level model may contain second level modules containing second level models and so on.

Clear, there is nothing like a „top“ level in universe (in UPS), Hence the top level must be identified by gravitational field instead of velocity and position. This is ffs.

Geographic Infrastructure, Tiles

The relations among modules, geographic infrastructure and tiles are ffs.

Identity, Virtual Identity, Real Identity

Need not be defined. If we need to explain this, then we do really have a problem.

Synchronization

SrrTrains uses the Network Sensor / Event Stream Sensor for synchronization of scene instances.

The objects that are used within PSIs and within the SCSI to synchronize the multiuser session, are called MIDAS Objects (Multiuser Interactivity Driven Animation and Simulation Objects).

The SCSI can be seen as relay between MIDAS Objects and the Internet of Things (IoT), where we define POIs (see below) as the peers of the SCSI, when it relays the IoT into the multiuser session.

Point of Interaction, Point of Interest (POI)

A POI is a unit that can be addressed within the IoT.

A Point of Interest delivers a stream of events to the multiuser session. This stream describes (a part of) the state of one or more RLOs.

A Point of Interaction accepts a stream of events from the multiuser session. This stream influences (a part of) the state of one or more RLOs.

A Point of Interaction may deliver a stream of events to the multiuser session. This stream describes (a part of) the state of one or more RLOs.

Connectivity Platform (CP)

X3D scenes communicate with Collaboration Servers (CS) through Network Sensors / Event Stream Sensors. CP is a conceptual name for an evolved CS.