

TorUsNet X ENG

Directional Architecture with a Self-Mutating Private Torus

The Architech (Gabriel Breton-Breault)

Introduction

TorUsNet X represents a new stage in the development of directional architecture. This evolution preserves the foundations established in TorUsNet 9.0, including the volumetric Private Torus, internal IDs, normal/ghost states, directional movements, and mutation mechanisms.

The fundamental difference in TorUsNet X, however, lies in the very nature of mutation.

In previous versions, the directional path could modify various internal states of the torus. TorUsNet X introduces an additional mechanism in which **the cells themselves can permute as the Private Torus evolves**.

The Private Torus is therefore no longer merely a structure in which internal data changes. Its own directional organization can be rearranged between two paths.

The general principle becomes:

$$T_n \xrightarrow{\text{path}} T_{n+1}$$

with:

$$T_n \neq T_{n+1}$$

The torus therefore possesses several successive configurations. Each new state results from the previous path and from the transformations determined by the cells encountered.

IDs as a Selection Mechanism

TorUsNet X uses the **IDs of the traversed cells** to determine which cells must participate in the permutation process.

In the previous architecture, IDs are represented using 16 bits and constitute directional markers associated with the cells of the Private Torus.

In TorUsNet X, reading these IDs acquires a new function.

The IDs encountered during the path are read alternately:

$$\begin{array}{l} ID_1 \rightarrow \\ ID_2 \leftarrow \\ ID_3 \rightarrow \\ ID_4 \leftarrow \end{array}$$

The first ID is read in its normal order.

The second is read in reverse order.

The third returns to normal reading.

The fourth is reversed again.

This alternation continues throughout the entire path.

When the resulting value is **odd**, the corresponding cell becomes a candidate for permutation:

$$\boxed{R(ID_i) \bmod 2 = 1 \Rightarrow C_i \in S}$$

where S represents the set of selected cells.

Selection therefore depends directly on **the directional content encountered during the path**.

Permutation of the Complete Cell

Permutation does not concern only the normal bit and the ghost bit.

In TorUsNet X, a cell must be considered as a complete directional structure.

When one cell permutes with another, the properties belonging to that cell move with it, including its ID and its associated directional properties.

The mechanism can be represented simply as:

$$\boxed{C_A \leftrightarrow C_B}$$

After permutation, the structure that occupied position A moves to position B , while the structure previously occupying position B takes position A .

The torus therefore undergoes an actual change in configuration.

$$\boxed{\text{State Mutation} + \text{Cell Permutation} = \text{Structural Mutation of the Torus}}$$

A Continuously Evolving Private Torus

This mechanism introduces a central characteristic of TorUsNet X:

The path transforms the torus that it has just traversed.

The Private Torus used for a given path is therefore not necessarily identical to the one that will be used for the next path.

The general sequence becomes:

$$\boxed{\text{Reading} \rightarrow \text{ID} \rightarrow \text{Selection} \rightarrow \text{Inversion} \rightarrow \text{Permutation} \rightarrow \text{New Torus State}}$$

Each path therefore contributes to the construction of the next configuration.

Two Temporary States for Synchronization

TorUsNet X also introduces a new synchronization logic between the Client and the Host.

The Client first applies the path to its Private Torus and generates:

$$T_{n+1}$$

During this operation, the Host temporarily retains the previous state:

$$T_n$$

while also preparing the new state:

$$T_{n+1}$$

The Host therefore temporarily maintains two states:

$$\boxed{T_n + T_{n+1}}$$

When the Client confirms that the path has been correctly applied, the Host can validate the transition and perform the switch:

$$\boxed{T_n \rightarrow T_{n+1}}$$

The new state then becomes the active Private Torus used to continue the session.

This evolution extends the synchronization principle already established in TorUsNet 9.0, in which Host and Client perform coordinated steps to maintain consistency between their Private Tori.

A Self-Rewriting Architecture

TorUsNet X can therefore be represented by the following sequence:

$T_n \rightarrow \text{Path} \rightarrow \text{Alternating ID Reading} \rightarrow \text{Cell Selection} \rightarrow \text{Inversion} \rightarrow \text{Permutation} \rightarrow T_{n+1}$

followed by:

$\text{Client Confirmation} \rightarrow \text{Host Validation} \rightarrow T_{n+1} \text{ Active}$

TorUsNet X therefore introduces the concept of a **self-mutating Private Torus**, whose directional organization evolves through its use.

The path no longer merely traverses the torus.

It participates in its transformation.

And each transformation prepares the directional structure within which the next path will evolve.

Chapter 1: The Structure of the Private Torus

1.1 Fundamental Volumetric Structure

In **TorUsNet X**, the Private Torus retains the volumetric architecture introduced in TorUsNet 9.0:

$100 \times 100 \times 100$

It therefore contains:

$1,000,000 \text{ cells}$

organized as:

$$\boxed{100 \text{ canvases of } 100 \times 100}$$

Each canvas contains:

$$100 \times 100 = 10,000 \text{ cells}$$

and constitutes a **directional layer** of the Private Torus.

The layers are arranged successively in depth:

$$C_1, C_2, C_3, \dots, C_{100}$$

This representation allows the Private Torus to be read along several different axes depending on the type of operation being performed.

The same torus can therefore be read:

- horizontally;
- vertically;
- circularly;
- in depth.

The reading geometry depends on the movement being executed.

1.2 The 100 Canvases as Successive Layers

The 100 canvases form the 100 successive levels of the Private Torus.

Each layer contains:

$$10,000 \text{ cells}$$

and the complete structure therefore contains:

$$100 \times 10,000 = 1,000,000 \text{ cells}$$

The structure can be represented conceptually as:

$$\boxed{C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow \dots \rightarrow C_{100}}$$

This succession does not, however, create an absolute physical boundary.

The Private Torus is a closed structure.

When a reading exceeds a boundary, it loops back according to the geometry corresponding to the movement being executed.

1.3 Circular Reading During an Exponential Jump

When an **exponential jump** is followed by a mathematical operation:

$$- + / *$$

the canvas is interpreted as **a circle**.

Reading begins from the cell determined as the starting point.

It then progresses in the direction imposed by the path.

If the reading reaches the final cell of the canvas and must continue, it does not stop.

It automatically returns to the beginning:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow 10,000 \rightarrow 1$$

The canvas therefore becomes a circular reading structure.

From the perspective of the path, there is no longer an absolute beginning and end.

There is only:

a starting point and a reading direction

1.4 Cardinal Movement and Toroidal Behavior

When a cardinal movement is executed, the canvas no longer behaves simply as a sequence of cells.

It behaves as **a torus**.

The opposite boundaries of the canvas become connected.

Thus:

top → bottom

and:

bottom → top

Likewise:

left → right

and:

right → left

A movement therefore never encounters an absolute boundary.

1.5 Vertical Reading in Columns

When a movement requires progression upward or downward, the reading mode changes.

Instead of reading the canvas horizontally, the Private Torus reads the cells **in columns, following the direction of movement.**

For a downward movement:

top ↓↓↓ bottom

The reading moves downward through the column.

When it exceeds the final position:

bottom → top

and reading continues.

For an upward movement, the principle is reversed:

bottom ↑↑↑ top

followed by:

top → bottom

The column therefore also forms **a circular loop**.

The direction of movement determines the reading direction.

1.6 Horizontal Reading

The same principle applies to movements toward the left and right.

To the right:

1 → 2 → 3 → ... → 100 → 1

To the left:

100 → 99 → 98 → ... → 1 → 100

Each row can therefore be considered a directional loop.

Each column can also be considered a directional loop.

The canvas therefore simultaneously possesses horizontal and vertical continuity.

1.7 Intensity and Depth Reading

TorUsNet X also retains the principle of **directional intensity**.

When an intensity is encountered, reading is no longer limited to the current canvas.

It progresses **in depth through the different layers of the Private Torus**.

The 100 canvases then become a vertical structure:

$C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow \dots \rightarrow C_{100}$

The intensity determines the depth reached by the reading.

1.8 Example of Positive Intensity

Consider:

$$I = 150$$

The Private Torus contains only 100 layers.

The reading therefore traverses the first 100 layers, loops back, and continues to layer 50.

The final layer is obtained through:

$$150 - 100 = 50$$

Thus:

$$I = 150 \Rightarrow C_{50}$$

The principle is therefore circular in depth:

$$C_{100} \rightarrow C_1$$

just as rows and columns loop within a canvas.

1.9 Bit Inversion During Traversal

When an intensity causes a traversal in depth, the traversed cells also undergo transformation.

Each encountered bit is inverted:

$$1 \rightarrow 0$$

and:

$$0 \rightarrow 1$$

Reading is therefore not passive.

The passage of the path modifies the state of the cells it traverses.

In TorUsNet X, this mechanism becomes one component of a broader mutation system, which also introduces **structural cell permutation determined by their IDs**.

The general transformation therefore becomes:

$$\text{Reading} \rightarrow \text{Traversal} \rightarrow \text{Bit Inversion}$$

before the additional TorUsNet X mechanisms are applied.

1.10 Negative Intensity

TorUsNet X also introduces a distinction based on the sign of the intensity.

A positive intensity produces a reading directed from the surface toward depth:

$$C_1 \rightarrow C_2 \rightarrow \dots \rightarrow C_{100}$$

A negative intensity produces the opposite reading direction.

Reading then begins from the deepest layers and moves toward the surface:

$$C_{100} \rightarrow C_{99} \rightarrow C_{98} \rightarrow \dots \rightarrow C_1$$

The sign of the intensity therefore becomes a **depth-direction operator**.

This rule can be represented as:

$$I > 0 \Rightarrow \text{surface} \rightarrow \text{depth}$$

and:

$$I < 0 \Rightarrow \text{depth} \rightarrow \text{surface}$$

The absolute value of the intensity determines the distance traveled:

$$|I|$$

while its sign:

$$\text{sgn}(I)$$

determines the reading direction.

1.11 Four Fundamental Reading Modes

The structure of the TorUsNet X Private Torus can therefore be understood through four mechanisms.

Circular Reading

Triggered by an exponential jump followed by a mathematical operation:

$$\text{Exponential Jump} + (-, +, /, *) \rightarrow \text{Circular Reading}$$

Toroidal Reading

Triggered by a cardinal movement:

$$\text{Cardinal Movement} \rightarrow \text{Toroidal Continuity}$$

Column Reading

Triggered when the movement is vertical:

$$\uparrow/\downarrow \rightarrow \text{Circular Column Reading}$$

Depth Reading

Triggered by an intensity:

$$I \rightarrow \text{Layer Reading}$$

with:

$$I > 0 \Rightarrow \text{surface} \rightarrow \text{depth}$$

and:

$$I < 0 \Rightarrow \text{depth} \rightarrow \text{surface}$$

1.12 General Principle of the Private Torus

The TorUsNet X Private Torus therefore constitutes a closed volumetric structure in which **each axis possesses its own continuity**.

There is no absolute boundary.

Horizontally, the reading loops.

Vertically, the reading loops.

In depth, the reading loops.

A path can therefore evolve along:

$\begin{matrix} \boxed{X} \\ \boxed{Y} \end{matrix}$

and:

$\boxed{Z}$

without leaving the volumetric structure.

The general principle can ultimately be summarized as:

$100 \times 100 \times 100 \rightarrow \text{Circular Reading} + \text{Toroidal Reading} + \text{Vertical Reading} + \text{Depth Reading}$

Intensity determines depth and its direction.

Cardinal movements determine local orientation.

Exponential jumps determine circular paths.

The traversed cells have their bits inverted.

TorUsNet X therefore retains the volumetric Private Torus of the previous generation while preparing its fundamental transformation:

The Private Torus is no longer merely a structure being traversed. Each path can modify the state in which the next path will encounter that structure.

Chapter 2: Cell Logic

2.1 Fundamental Structure of a Cell

In **TorUsNet X**, each cell of the Private Torus constitutes a complete directional unit.

Each cell contains:

- a **16-bit ID**;
- a **normal bit**;

- a **ghost bit**;
- **four possible movement cells**;
- an **exponential jump** associated with a mathematical operation;
- a **cardinal direction**;
- a **private intensity**, either positive or negative.

A cell can therefore be conceptually represented as:

$$C_i = (I, D_i, B_i, \bar{B}_i, D_{i1}, D_{i2}, D_{i3}, D_{i4} S_i O_i D_i I_i)$$

The cell is therefore not merely a position within a grid. It contains the information that allows the path to determine **what to read, how to move, and how to progressively transform the Private Torus**.

2.2 The Normal Bit and the Ghost Bit

Each cell contains a normal bit:

$$B_i \in \{0, 1\}$$

and a ghost bit that always represents its opposite:

$$\bar{B}_i = NOT(B_i)$$

Thus:

$$B_i = 0 \Rightarrow \bar{B}_i = 1$$

and:

$$B_i = 1 \Rightarrow \bar{B}_i = 0$$

The two states therefore form a complementary pair.

When an inversion is triggered:

$$0 \leftrightarrow 1$$

the ghost bit automatically remains the opposite of the new normal state.

2.3 The Four Movement Possibilities

Each cell possesses four possible movements.

These movements allow the path to determine the next cell that can be reached from the current cell.

The choice of movement depends on the directional logic applied to the path.

The current cell therefore becomes a transition between several potential positions:

$$C_i \rightarrow \{C_a, C_b, C_c, C_d\}$$

The cell therefore does not represent only information to be read.

It also represents **a set of possible paths**.

2.4 The Exponential Jump and Mathematical Operation

Each cell also possesses an exponential jump associated with one of the following operations:

$$\boxed{+} \boxed{-} \boxed{/} \boxed{*}$$

When an exponential jump followed by one of these operations is executed, reading follows the circular logic defined in the previous chapter.

The path can therefore move beyond the apparent boundary of the canvas and return to the beginning.

The exponential jump thus constitutes **an internal movement mechanism of the cell**.

2.5 Cardinal Direction

A cell also possesses a cardinal direction.

This direction determines how the path must interpret movement within the canvas.

Depending on the direction, the path can progress:

$$\boxed{\uparrow} \boxed{\downarrow} \boxed{\leftarrow} \boxed{\rightarrow}$$

The toroidal geometry then imposes the continuity described in Chapter 1.

If the path exceeds one boundary of the canvas, it reappears on the opposite side.

2.6 The Cell's Private Intensity

Each cell also contains its own intensity:

$$I_{\text{private}}$$

This intensity can be:

$$I_{\text{private}} > 0$$

or:

$$I_{\text{private}} < 0$$

It is added to the intensity transmitted by the path over the network:

$$I_{\text{final}} = I_{\text{transmitted}} + I_{\text{private}}$$

The final intensity then determines the depth reading of the Private Torus.

The sign of this intensity also determines its orientation:

$$I_{\text{final}} > 0 \Rightarrow \text{surface} \rightarrow \text{depth}$$

$$I_{\text{final}} < 0 \Rightarrow \text{depth} \rightarrow \text{surface}$$

Intensity therefore becomes a dynamic property constructed from a transmitted component and a component belonging to the Private Torus.

2.7 Alternating ID Reading

TorUsNet X introduces a new fundamental function: **alternating reading of the IDs of traversed cells**.

Each ID contains:

$$16 \text{ bits}$$

The reading direction depends on the position of the cell within the path.

The first ID is read normally:

$$ID_A \rightarrow$$

The second ID is read in reverse:

$$ID_B \leftarrow$$

The third returns to normal reading:

$$ID_C \rightarrow$$

The fourth is read in reverse:

$$ID_D \leftarrow$$

and so on.

The general rule becomes:

$$R(ID_i) = \begin{cases} ID_i \text{ read normally,} & i \text{ odd} \\ ID_i \text{ read in reverse,} & i \text{ even} \end{cases}$$

This alternation continues throughout the entire path.

2.8 Parity as a Permutation Trigger

After an ID is read, TorUsNet X checks the parity of the resulting value.

If:

$$R(ID_i) \bmod 2 = 0$$

the value is even, and no permutation is requested by that cell.

If:

$$R(ID_i) \bmod 2 = 1$$

the value is odd, and the cell must participate in the permutation mechanism.

The rule therefore becomes:

$$\boxed{ID_{\text{read}} \text{ odd} \Rightarrow \text{permutation}}$$

Parity therefore determines more than a logical value.

It becomes a **structural trigger of the Private Torus**.

2.9 Example: Reading Cell A

Consider the path:

$$A \rightarrow B \rightarrow C \rightarrow D$$

The path begins with cell A .

The ID of A is the first ID of the path. It is therefore read normally:

$$ID_A \rightarrow$$

If the resulting value is odd:

$$R(ID_A) 2 = 1$$

a permutation is scheduled between:

$$\boxed{A \leftrightarrow B}$$

However, this permutation is **not executed immediately**.

The path first continues its progression.

It reads:

$$A \rightarrow B \rightarrow C$$

When cell C is reached, the previously scheduled permutation can be applied:

$$\boxed{A \leftrightarrow B}$$

The path therefore retains sufficient information to continue its execution before the two previous cells change position.

2.10 Example: Reading Cell B

Cell B constitutes the second cell of the path.

Its ID is therefore read in reverse:

$$ID_B \leftarrow$$

If the reversed value is odd:

$$R(ID_B) \bmod 2 = 1$$

a new permutation is recorded:

$$\boxed{B \leftrightarrow C}$$

This permutation also waits until the next cell has been read.

The path becomes:

$$A \rightarrow B \rightarrow C \rightarrow D$$

Then, when D is read:

$$\boxed{B \leftrightarrow C}$$

is applied.

The general principle therefore becomes:

The parity of a cell determines a permutation with the following cell, but the exchange is executed only after the subsequent cell has been read.

2.11 One-Cell Delay Before Permutation

This delay constitutes an important property of TorUsNet X.

If the current cell is:

$$C_i$$

and its interpreted ID is odd, TorUsNet X schedules:

$$C_i \leftrightarrow C_{i+1}$$

but waits for the reading of:

$$C_{i+2}$$

before executing the exchange.

Formally:

$$\boxed{R(ID_i) \text{ odd} \Rightarrow \text{schedule}(C_i \leftrightarrow C_{i+1})}$$

followed by:

$$\boxed{\text{read}(C_{i+2}) \Rightarrow \text{execute}(C_i \leftrightarrow C_{i+1})}$$

This allows the path to advance before the structure it has just traversed is reorganized.

2.12 Bit Inversion During the Process

Structural permutation does not replace bit inversion.

Both mechanisms coexist.

When a traversed cell must undergo an inversion:

$$\boxed{B_i \leftarrow NOT(B_i)}$$

Thus:

$$\boxed{1 \rightarrow 0}$$

and:

$$\boxed{0 \rightarrow 1}$$

The ghost bit automatically remains the opposite of the normal bit:

$$\boxed{\bar{B}_i = NOT(B_i)}$$

Therefore, when A and B permute, their logical states may also have been transformed during the path.

The process therefore simultaneously involves:

logical mutation

and:

structural mutation

2.13 The Entire Cell Participates in the Permutation

A permutation must not be interpreted merely as the exchange of two bits.

TorUsNet X considers the cell as a complete structure.

Thus:

$$A \leftrightarrow B$$

means that the structures associated with positions A and B are reorganized according to the protocol's permutation function.

This allows the Private Torus to modify not only its internal values, but also **the arrangement of paths available to future directional paths**.

The next path therefore encounters a Private Torus different from the one that existed before the previous path.

2.14 Complete Execution Sequence

For a sequence:

$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$$

the operation can be summarized as follows.

Reading A

$$ID_A \rightarrow$$

If odd:

$$\boxed{\text{schedule } A \leftrightarrow B}$$

Reading B

$$ID_B \leftarrow$$

If odd:

$$\boxed{\text{schedule } B \leftrightarrow C}$$

Reading C

The permutation previously triggered by A can be applied:

$$\boxed{A \leftrightarrow B}$$

Reading D

The permutation previously triggered by B can be applied:

$$\boxed{B \leftrightarrow C}$$

The mechanism then continues according to the same logic for the following cells.

2.15 General Principle of Cell Logic

In TorUsNet X, a cell therefore combines several functions:

$$\boxed{\text{Cell} = \text{ID} + \text{Bits} + \text{Movements} + \text{Jump} + \text{Operation} + \text{Direction} + \text{Intensity}}$$

The ID participates in permutation selection.

The bits represent the normal/ghost logical state.

The movements define the possible paths.

The exponential jump determines circular progression.

The cardinal direction defines local orientation.

Intensity determines depth reading.

And permutation modifies the organization of the Private Torus after the path has passed through it.

The general cycle becomes:

Cell Reading → Alternating ID Reading → Parity Test → Movement → Next Cell Reading → Bit Inversion →
--

Thus:

A TorUsNet X cell is no longer merely a location containing information. It is an active directional unit capable of participating in the transformation of the Private Torus itself.

The path therefore leaves behind a structure different from the one it encountered:

$$\boxed{T_n \xrightarrow{\text{Cell Reading}} T_{n+1}}$$

It is this progressive transformation that makes the **TorUsNet X** Private Torus a truly **self-mutating** structure.

Chapter 3: The Path Transmitted Over the Network

3.1 General Principle

In **TorUsNet X**, the directional path is transmitted over the network so that the Host and the Client can apply the same reading sequence to their respective Private Tori.

The process begins with a negotiation that establishes the starting cell and the destination cell. Once this negotiation is complete, the path can be transmitted and executed.

The general sequence is:

Negotiation → Validation → Path Transmission → Reconstruction → Mutation → Synchronization
--

3.2 The Initial Key

Before transmitting the path, the Host generates a key containing the information required for its initialization.

This key contains:

- the **ID of the starting cell**;
- the **destination cell**.

It can be conceptually represented as:

$$K = ID_{\text{start}} + C_{\text{destination}}$$

This key does not contain the complete path.

It allows the Host and the Client to establish the same reference point before executing the path.

3.3 Negotiation Between the Host and the Client

Negotiation must occur before the path is transmitted.

First Transmission

The Host transmits the key to the Client:

$$Host \xrightarrow{K} Client$$

The Client receives and verifies the key.

Second Transmission

The Client returns the same key to the Host:

$$Client \xrightarrow{K} Host$$

This operation confirms that the Client received the correct key.

Third Transmission

The Host sends the same key back to the Client once again to confirm that it recognizes the returned key:

$$Host \xrightarrow{K} Client$$

Confirmation

Receipt of this final key is confirmed.

From this point onward:

The path can be transmitted

This negotiation therefore establishes a common state between both systems before execution of the path.

3.4 Structure of the Transmitted Path

In TorUsNet X, each bit reconstructed from the path corresponds to an instruction transmitted using:

11 bits

These 11 bits are divided into three fields:

DD | *IIIIIII* | *B*

with:

- **2 bits** to select the movement;
- **8 bits** for the transmitted intensity;
- **1 bit** to select the normal bit or the ghost bit.

Thus:

$$2 + 8 + 1 = 11 \text{ bits}$$

3.5 The Two Movement Bits

The first two bits indicate which of the four movement possibilities contained within the cell must be used.

The four codes are:

00

01

10

11

Each combination identifies one of the four available movement cells.

This operation can be represented as:

$$DD \rightarrow D_{i,DD}$$

The path therefore does not need to directly transmit all the properties of the next cell.

It selects one of the four movements already present within the structure of the Private Torus.

3.6 Intensity Transmission

After the two movement bits, the path transmits an intensity using:

$$8 \text{ bits}$$

This value constitutes the transmitted intensity:

$$I_T$$

It is then combined with the intensity belonging to the selected movement.

3.7 Special Case of the Starting Cell

For the starting cell, the transmitted intensity is used directly.

There is therefore no addition with a previous movement intensity.

For the first cell:

$$I_F = I_T$$

where:

- I_T = transmitted intensity;
- I_F = final intensity used.

The starting cell therefore initializes the directional behavior of the path.

3.8 Adding the Movement Intensity

After the starting cell, the selected movement possesses its own intensity.

This intensity can be:

$$I_P > 0$$

or:

$$I_P < 0$$

It is added to the transmitted intensity:

$$I_F = I_T + I_P$$

The final intensity I_F then becomes the value actually used for depth reading within the Private Torus.

3.9 Positive or Negative Intensity

The sign of the final intensity determines the direction of depth reading.

When:

$$I_F > 0$$

reading progresses from the surface toward depth:

$$C_1 \rightarrow C_2 \rightarrow \cdots \rightarrow C_{100}$$

When:

$$I_F < 0$$

the direction is reversed:

$$C_{100} \rightarrow C_{99} \rightarrow \cdots \rightarrow C_1$$

The value of the intensity therefore determines progression, while its sign determines the direction of that progression.

3.10 Mutation of the Movement Intensity

The addition:

$$I_F = I_T + I_P$$

does not merely determine the current reading.

It also transforms the Private Torus.

After the operation, the previous intensity of the selected movement is replaced by the resulting sum:

$$I_P^{new} = I_F$$

Thus:

$$I_P \rightarrow I_T + I_P$$

The next path encountering this same movement will therefore encounter a new intensity.

The future behavior of the Private Torus consequently depends on paths that have already been executed.

3.11 Selection of the Normal or Ghost Bit

The final bit of the 11-bit instruction determines which type of bit must be read.

It selects:

Normal Bit

or:

Ghost Bit

The ghost bit always remains the opposite of the normal bit:

$$B_F = NOT(B_N)$$

The final bit of the instruction therefore indicates which manifestation of the cell must participate in the reconstruction.

3.12 Reconstruction of a Bit

Each transmitted group of 11 bits therefore produces a complete operation:

$$DD + I_T + B$$

The Private Torus:

1. selects one of the four movements;
2. retrieves the transmitted intensity;
3. combines this intensity with the private intensity of the movement, except for the starting cell;
4. determines the depth reading;
5. updates the movement intensity;
6. reaches the corresponding cell;
7. selects the normal or ghost bit;
8. reconstructs the corresponding bit.

Each reconstructed bit therefore requires:

$$11 \text{ transmitted path bits}$$

3.13 The First 16 Bits of the Reconstructed File

TorUsNet X introduces an additional function for the **first 16 bits of the reconstructed file**.

These bits normally belong to the file.

They do not constitute an additional block.

They are not removed from the file after being read.

They simply possess **a second simultaneous function**.

If the reconstructed file begins with:

$$F = b_1 b_2 b_3 \dots b_{16} b_{17} \dots b_n$$

then:

$$b_1 b_2 b_3 \dots b_{16}$$

constitute both:

1. the first 16 bits of the file;
2. the new ID of the starting cell.

Thus:

$$ID_{\text{start}}^{n+1} = b_1 b_2 b_3 \dots b_{16}$$

3.14 Dual Function of the First 16 Bits

The fundamental rule is therefore:

$$B_{1 \rightarrow 16} \rightarrow \begin{cases} \text{Beginning of the reconstructed file} \\ \text{New starting-cell ID} \end{cases}$$

The file remains complete:

$$F = b_1 b_2 \dots b_{16} b_{17} \dots b_n$$

No bit is removed.

The ID mutation simply uses information that has already been reconstructed.

The file therefore directly participates in the evolution of the Private Torus.

3.15 Mutation of the Starting ID

Before the path is executed, the starting cell possesses:

$$ID_{\text{start}}^n$$

After reconstruction of the first 16 bits:

$$ID_{\text{start}}^{n+1} = b_1 b_2 \dots b_{16}$$

The mutation becomes:

$$ID_{\text{start}}^n \rightarrow ID_{\text{start}}^{n+1}$$

The new ID belongs to the **new state of the Private Torus**.

It therefore does not necessarily immediately replace the ID belonging to the currently active state.

It is prepared for:

$$T_{n+1}$$

3.16 Interaction with the Permutation Logic

While the file is being reconstructed, the other TorUsNet X mechanisms continue operating.

The encountered IDs are read alternately:

$$\rightarrow, \leftarrow, \rightarrow, \leftarrow, \dots$$

Odd values can schedule permutations between cells.

Traversed bits can change state:

$$0 \leftrightarrow 1$$

Intensities evolve:

$$I_p^{new} = I_T + I_P$$

and cells scheduled for permutation change position according to the rules defined in the previous chapter.

The same path therefore simultaneously produces several transformations.

3.17 Two Results of the Path

The transmitted path therefore has two fundamental results.

Reconstruction

$$P \rightarrow F$$

The path produces the reconstructed file.

Evolution of the Private Torus

$$T_n \xrightarrow{P} T_{n+1}$$

The path also produces the next configuration of the torus.

This new configuration notably includes:

- the new ID of the starting cell;
- the new intensities;
- the new bit states;
- cell permutations;
- the resulting new directional organization.

Thus:

$$\text{One Path} \rightarrow \text{One File} + \text{One New Torus State}$$

3.18 Synchronization of the New State

After execution of the path, the Client possesses the new state:

$$T_C^{n+1}$$

The Host temporarily retains:

$$T_H^n$$

and prepares:

$$T_H^{n+1}$$

The Host can therefore temporarily maintain:

$$T_H^n + T_H^{n+1}$$

The Client then confirms that the path has been applied.

After validation, the Host can perform the switch:

$$T_H^n \rightarrow T_H^{n+1}$$

The new state then becomes active.

The ID produced from the first 16 bits becomes active with it:

$$ID_{\text{start}}^n \rightarrow ID_{\text{start}}^{n+1}$$

The Host and the Client then continue operating from the synchronized new state.

3.19 Complete Structure of a Network Instruction

A path instruction therefore retains the structure:

$$DD \mid IIIIIII \mid B$$

or:

$$2 + 8 + 1 = 11 \text{ bits}$$

The corresponding functions are:

$$DD \rightarrow \text{Movement}$$

$$IIIIIII \rightarrow I_T$$

$$B \rightarrow \text{Normal/Ghost}$$

Then:

$$I_F = I_T + I_P$$

except for the initial cell, for which:

$$I_F = I_T$$

After the operation:

$$I_P^{new} = I_F$$

3.20 General Principle of the Transmitted Path

The path transmitted over the network therefore becomes much more than a simple list of movements.

It simultaneously constitutes:

- a movement instruction;
- an intensity instruction;
- a reading instruction;
- a reconstruction mechanism;
- an intensity mutation mechanism;
- an ID mutation mechanism;
- an inversion trigger;
- a cell permutation engine;
- a mechanism that produces the next state of the Private Torus.

The general sequence becomes:

Negotiation → Network Path → Movement → Intensity → Normal/Ghost Reading → Reconstruction → M

And one of the distinctive characteristics of TorUsNet X becomes:

First 16 Bits of the File = Content + New ID

The path therefore reconstructs the file while simultaneously participating in the transformation of the structure that has just reconstructed it.

In TorUsNet X, the transmitted path simultaneously produces information and a new directional state of the Private Torus.

General Conclusion

TorUsNet X represents a new stage in the evolution of directional architecture. This version preserves the foundations established in TorUsNet 9.0 while introducing a major evolution: the Private Torus is no longer merely a structure in which internal states change. **The structure itself evolves as it is used.**

The Private Torus remains based on a volumetric architecture:

$$100 \times 100 \times 100$$

or:

$$1,000,000 \text{ cells}$$

organized into one hundred successive 100×100 canvases.

This structure can be traversed horizontally, vertically, circularly, and in depth. Cardinal movements use the toroidal continuity of the canvases, while exponential jumps associated with the operations $+$, $-$, $/$, and $*$ allow circular reading from the starting point.

Intensity introduces a third reading direction: **depth**.

A positive intensity produces progression from the surface toward the deeper layers:

$$C_1 \rightarrow C_2 \rightarrow \dots \rightarrow C_{100}$$

while a negative intensity reverses this orientation:

$$C_{100} \rightarrow C_{99} \rightarrow \dots \rightarrow C_1$$

A Cell as an Active Directional Structure

In TorUsNet X, each cell constitutes a complete unit containing, among other elements:

- a 16-bit ID;
- a normal bit;
- a ghost bit that is always the opposite of the normal bit;
- four movement possibilities;

- exponential jumps associated with operations;
- a cardinal direction;
- a positive or negative intensity.

The cell is therefore no longer simply a location containing a value.

It directly participates in the progression and transformation of the Private Torus.

IDs are read alternately according to their position in the path:

$\rightarrow, \leftarrow, \rightarrow, \leftarrow, \dots$

When an ID interpreted in this way produces an odd value, the corresponding cell triggers a permutation with the following cell.

This permutation is delayed so that the path can first reach the subsequent cell.

The structure can therefore evolve according to a sequence such as:

$$A \rightarrow B \rightarrow C$$

followed by:

$A \leftrightarrow B$

The path therefore progressively modifies the arrangement of the structures it has just traversed.

The Transmitted Path

Communication between the Host and the Client begins with negotiation of the starting cell and destination cell.

After validation, the path is transmitted as instructions of:

11 bits

for each reconstructed bit:

2 movement bits + 8 intensity bits + 1 normal/ghost bit

or:

$$\boxed{DD \mid IIIIII \mid B}$$

The first two bits select one of the four movement possibilities.

The following eight bits transmit the intensity.

The final bit determines whether reconstruction must use the normal or ghost state of the cell.

Beginning with the second movement, the transmitted intensity is combined with the private intensity of the selected movement:

$$\boxed{I_F = I_T + I_P}$$

The private intensity of the movement then evolves itself:

$$\boxed{I_P^{new} = I_F}$$

Each path therefore contributes to modifying the conditions under which the next path will be executed.

The File Participates in the Mutation of the Torus

The first 16 bits of the reconstructed file possess a dual function.

They remain fully present within the file:

$$\boxed{F = b_1 b_2 \dots b_{16} b_{17} \dots b_n}$$

but they also produce the new ID of the starting cell:

$$\boxed{ID_{\text{start}}^{n+1} = b_1 b_2 \dots b_{16}}$$

Thus:

$$\boxed{\text{First 16 Bits} \rightarrow \begin{cases} \text{File Content} \\ \text{New Starting-Cell ID} \end{cases}}$$

The reconstructed file therefore directly contributes to the transformation of the structure that has just reconstructed it.

A Self-Mutating Private Torus

TorUsNet X ultimately leads to a fundamental relationship:

$$T_n \xrightarrow{\text{Path}} T_{n+1}$$

The new state can incorporate:

- inversion of the traversed bits;
- modification of intensities;
- mutation of the starting ID;
- cell permutations;
- reorganization of the associated directional properties.

The path therefore simultaneously serves to:

Reconstruct + Transform

The Host can temporarily retain both the previous and new states of the torus. When the Client confirms that the path has been correctly applied, the new state becomes the synchronized active state.

The general logic of TorUsNet X can therefore be summarized as:

Negotiation → Path → Reading → Reconstruction → Inversion → Mutation → Permutation → T_{n+1}

The central principle of **TorUsNet X** can ultimately be expressed as follows:

The path no longer simply traverses the Private Torus. It participates in the creation of its next configuration. The file is reconstructed while the bits, intensities, IDs, and directional organization of the torus evolve. Each path therefore leaves behind a new state intended for the next path.

Essential Companion: TorUsNet 9.0

TorUsNet X must be read in conjunction with TorUsNet 9.0.

This document focuses primarily on the new mechanisms introduced by TorUsNet X and is not intended to repeat the entire architecture already described in TorUsNet 9.0, particularly its volumetric foundations and the protocol mechanisms established in the previous generation.

To understand the complete evolution of the architecture, both documents should therefore be considered together:

TorUsNet 9.0 + TorUsNet X

TorUsNet 9.0 establishes the foundations.

TorUsNet X develops the principle of the **self-mutating Private Torus**.

Official Website

The publications, developments, and other works of **The Architech (Gabriel Breton-Breault)** are available at:

<https://Kyzis.org>

TorUsNet X is therefore part of the ongoing development of directional architectures published on Kyzis.org.

TorUsNet X: the path reconstructs the information, while also transforming the path that will make it possible to reconstruct the next information.