Forctis ARES Token

September 2018

Creating a **new** technology
for a **better** world

Example of a 20 cents token issued by Ingenio Ledesma, a sugar cane mill in northern Argentina (Jujuy). The usual pay for a Mataco (indigenous people of Wichí origin) was 12 token pesos per month plus food; rations were exchanged against rubber tokens given at the end of the day journey. See J. B. Massé. El estado de las clases obreras a comienzos del siglo. Cordoba: Universidad Nacional de Cordoba, 1968.

Welcome to an overview of ARES. The word is an acronym for Asset Representation System. The name highlights a key design objective behind the token that will become integral to our blockchain-inspired platform.

Please feel free to contact us if you require more information.

info@forctis.io

The ability to document property rights[1] is one of the key principles that underpins the functioning of modern economies.

It provides support to the notion of representation. That is precisely what ARES aims to facilitate for everyone in society.

[1] Ensuring, as a result, the right to own, fractionalize and to transfer property.

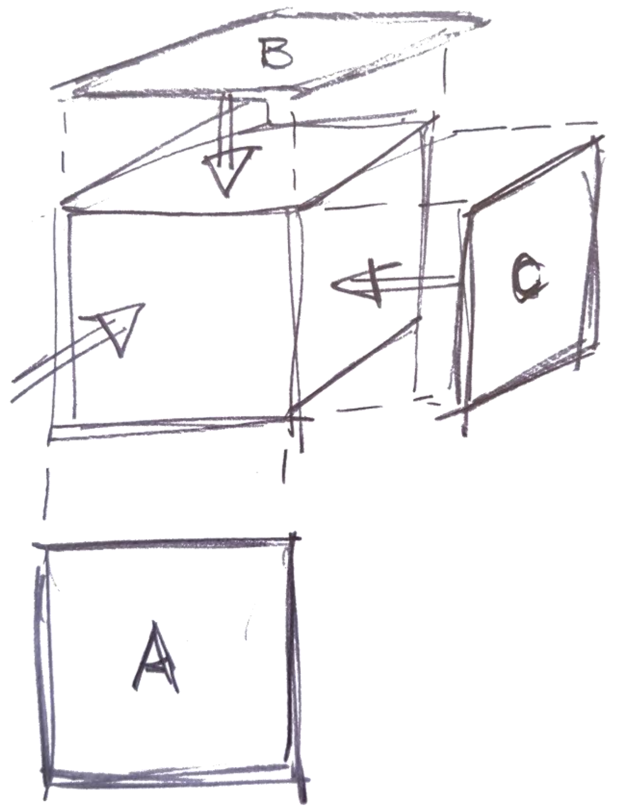
"Sorry to be a wet blanket. Writing a description for this thing for general audiences is bloody hard. There's nothing to relate it to."

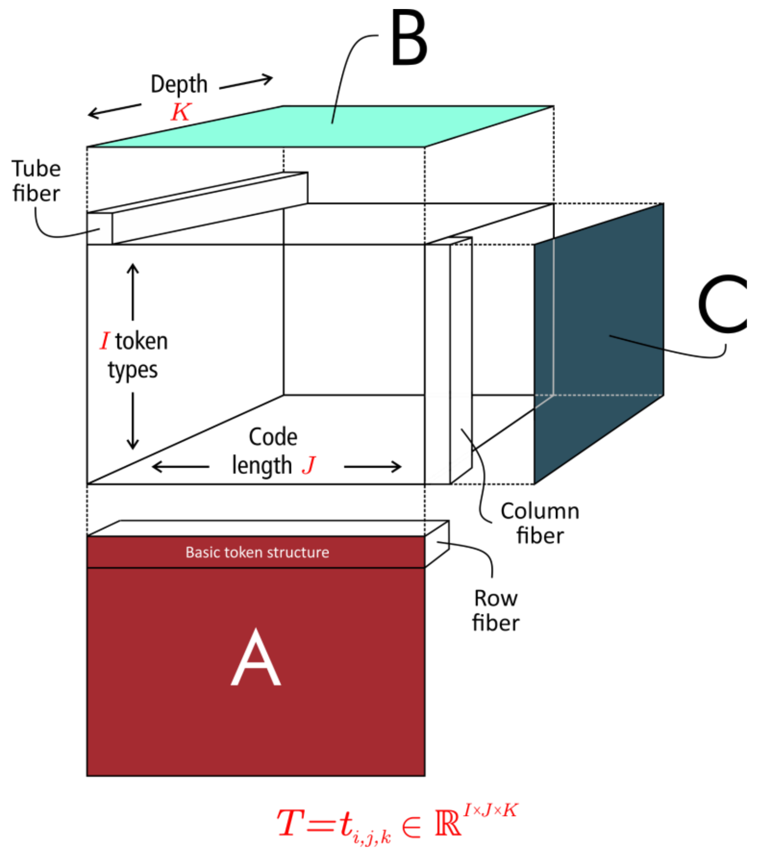
Satoshi Nakamoto on Bitcointalk (2009)
https://bitcointalk.org/index.php?topic=234.msg1976#msg1976

Anyway, let’s make a try.

Forctis

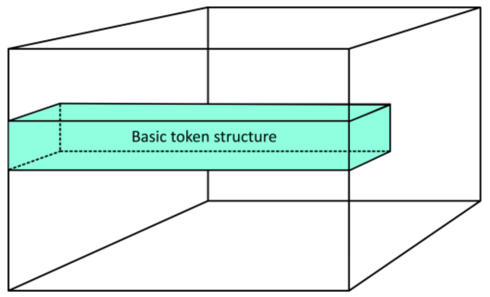
The original design sketch



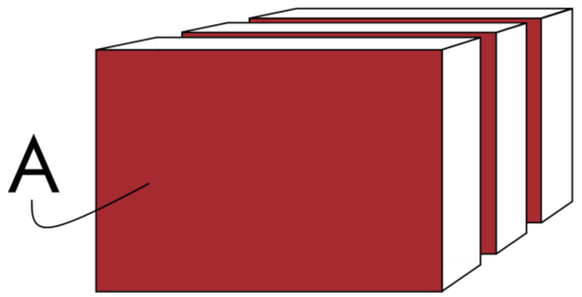


The ARES token resembles a cube. It was designed to provide a versatile structure for incorporating multiple asset classes. Each token is unique as unique are the assets that its users might wish to digitally represent into them.

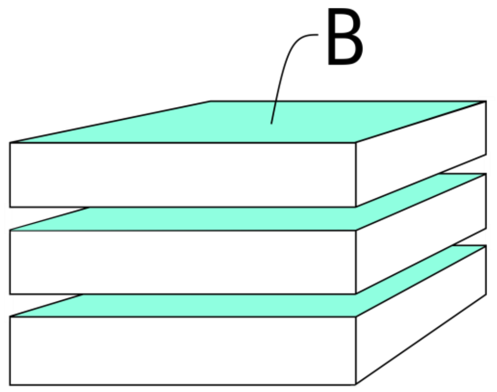
ARES can be partitioned in several ways, but our focus here is placed on two basic "geometrical" structures.[2]



Row fiber
(the basic functional structure of ARES)



Frontal slabs
(in the direction of the row fibers)



Horizontal slabs
(in the direction of the asset class)

Fibers

Slabs (or slices)

- **Row fiber**

  It defines the <u>basic functional token structure</u> of ARES.

- **Frontal slabs**

  Made by row fibers stacked one on top of the other. Each fiber represents <u>assets that belong to different asset classes</u>.

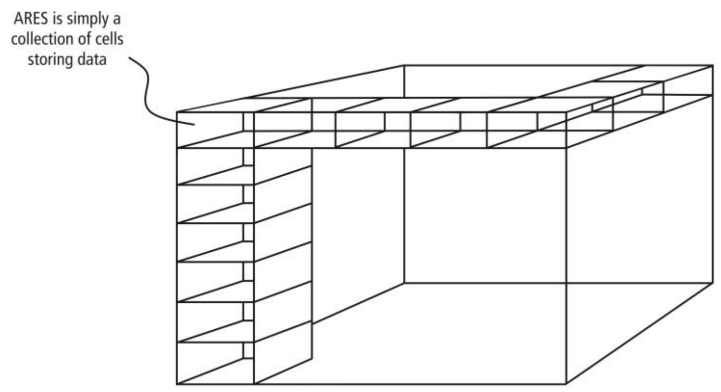
- **Horizontal slabs**

  They group different fibers alongside each other. Here, the fibers correspond to <u>different assets belonging to the same asset class</u>.

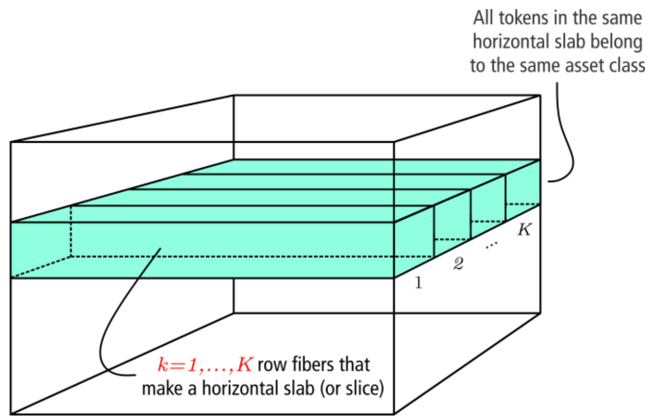
[2] It is worth noting that we speak of a "geometry" in order to facilitate visualization. ARES is basically a container of numeric high-order data. Compared to a "geometric" construct which must meet strict mathematical or physical properties, this is not necessarily the case for every use of numeric data. See, e.g. A. P. Harrison and D. Joseph. *Numeric Tensor Framework: Exploiting and Extending Einstein Notation.* Journal of Computational Science, 16 (2016), pp. 128-139.

Forctis

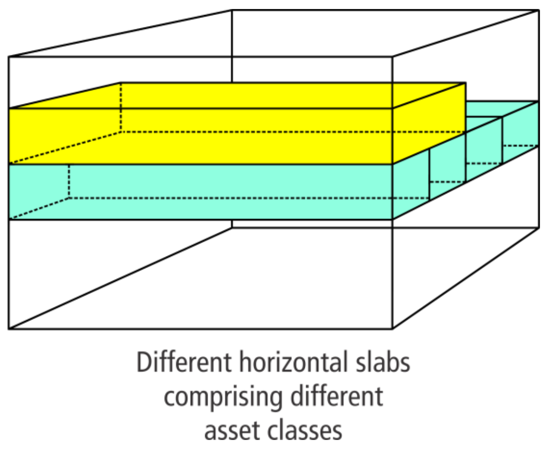
A



B

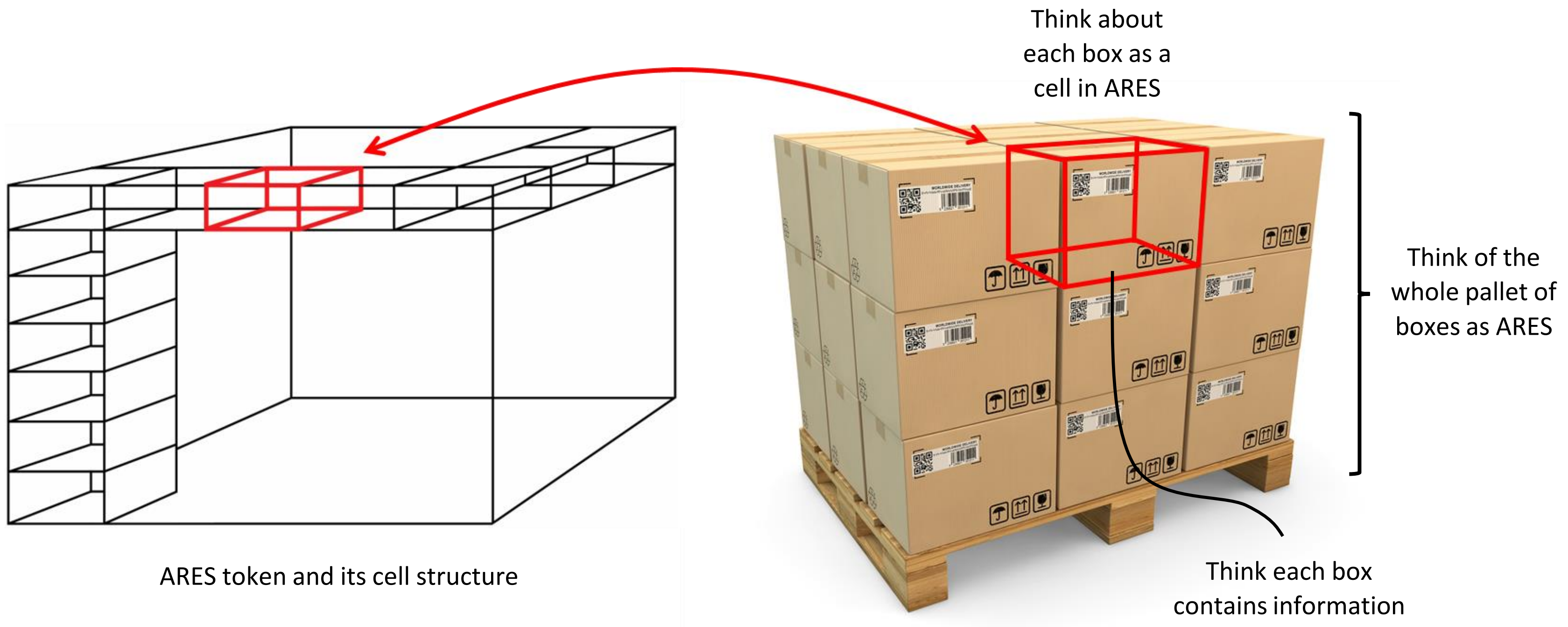

All tokens in the same horizontal slab belong to the same asset class
1
2
...
K
$k=1,\ldots,K$ row fibers that make a horizontal slab (or slice)

C


Different horizontal slabs comprising different asset classes

The data cells A of ARES store specific pieces of information. Those cells are the elementary building blocks of ARES. A row fiber is then built from a pre-determined number of cells. As shown before, row fibers B are the simplest functional representation of ARES. When the fibers representing multiple assets of the same class are grouped together they form a slab. Tokens (or slabs) having a different number of cells C can be embedded into ARES to create more complex structures.

Forctis



ARES token and its cell structure

A simple way of visualizing the cells of ARES (and ARES in more general terms) is to make an analogy with a pallet stacked with boxes. Think of ARES as all the boxes stacked on that pallet and the cells of ARES as each of those boxes.

Forctis

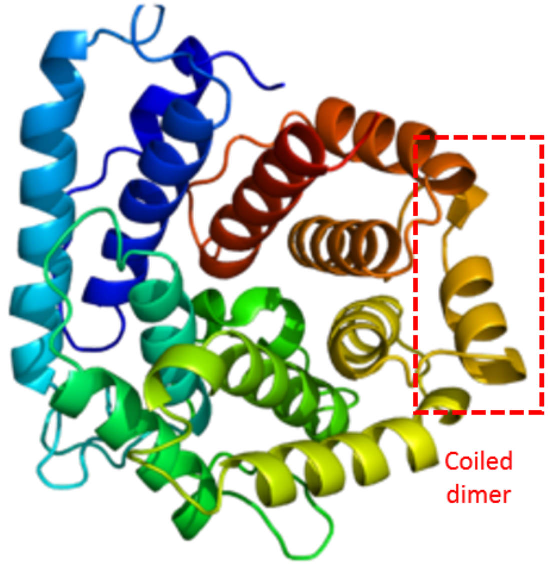
Going a bit further (what “inspires” our design)

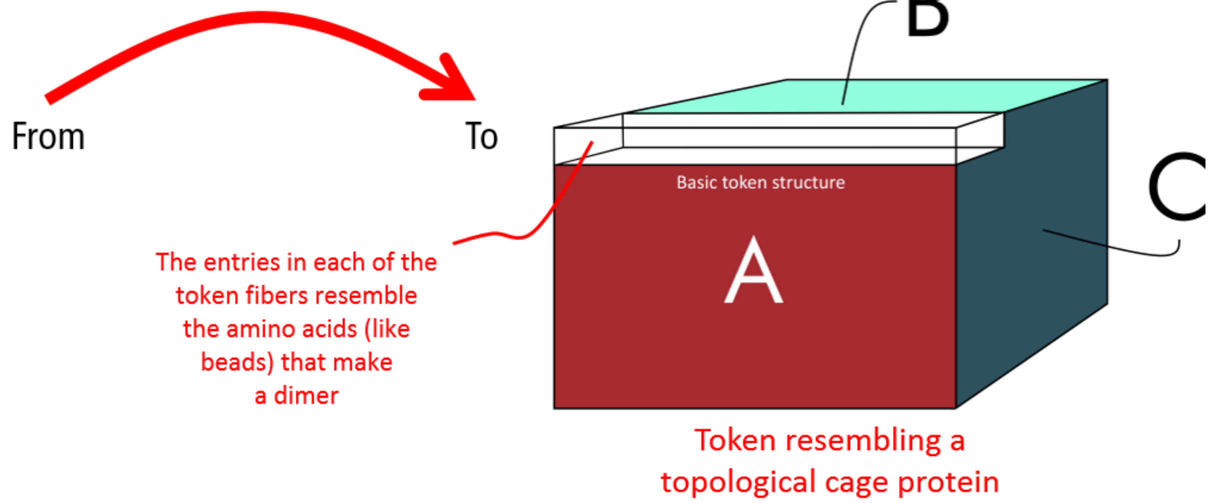
Molecular representation

The protein

Mathematical synthesis

ARES



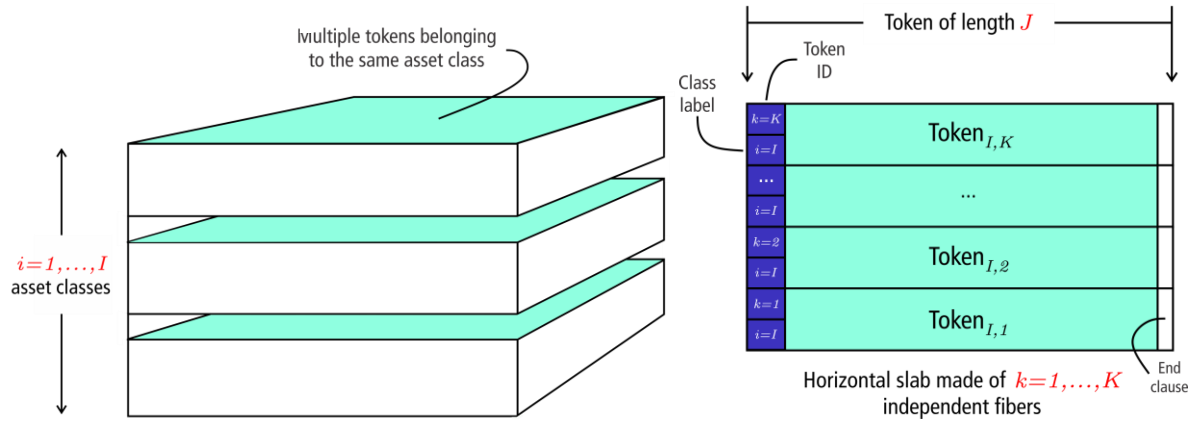

From
To
B
C
Basic token structure
A
The entries in each of the token fibers resemble the amino acids (like beads) that make a dimer
Token resembling a topological cage protein

ARES can be seen as the mathematical representation of a protein structure. Each fiber in the token is the equivalent of a coiled dimer. Like the dimers in a real-life protein, the token is composed of a sequence of topological folds (using those fibers as building modules) that are stacked together in a structure akin to a tensor.

Looking under the skin of ARES

Forctis

1 Horizontal slabs



Multiple tokens belonging to the same asset class
$i=1,\dots,I$ asset classes
Token of length J
Token ID
Class label
$k=K$
$i=I$
Token$_{I,K}$
...
$k=2$
Token$_{I,2}$
$k=1$
Token$_{I,1}$
End clause
Horizontal slab made of $k=1,\dots,K$ independent fibers

Examples when information requires $>J$ cells to be encoded

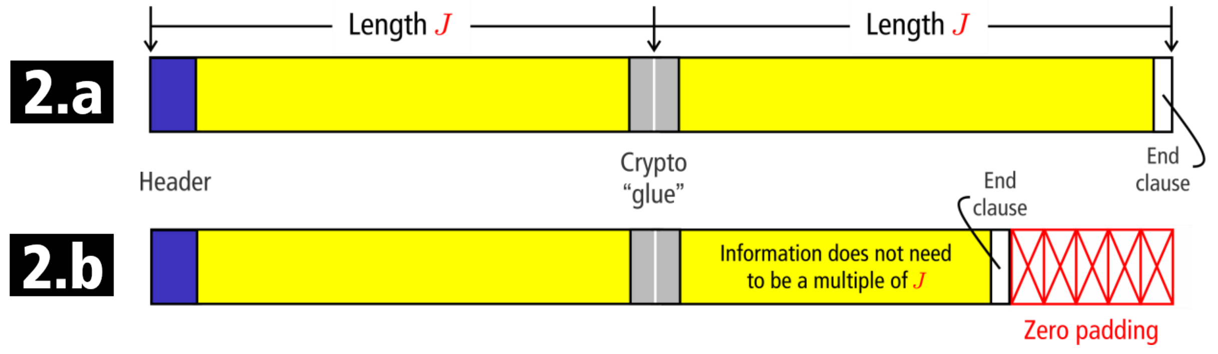
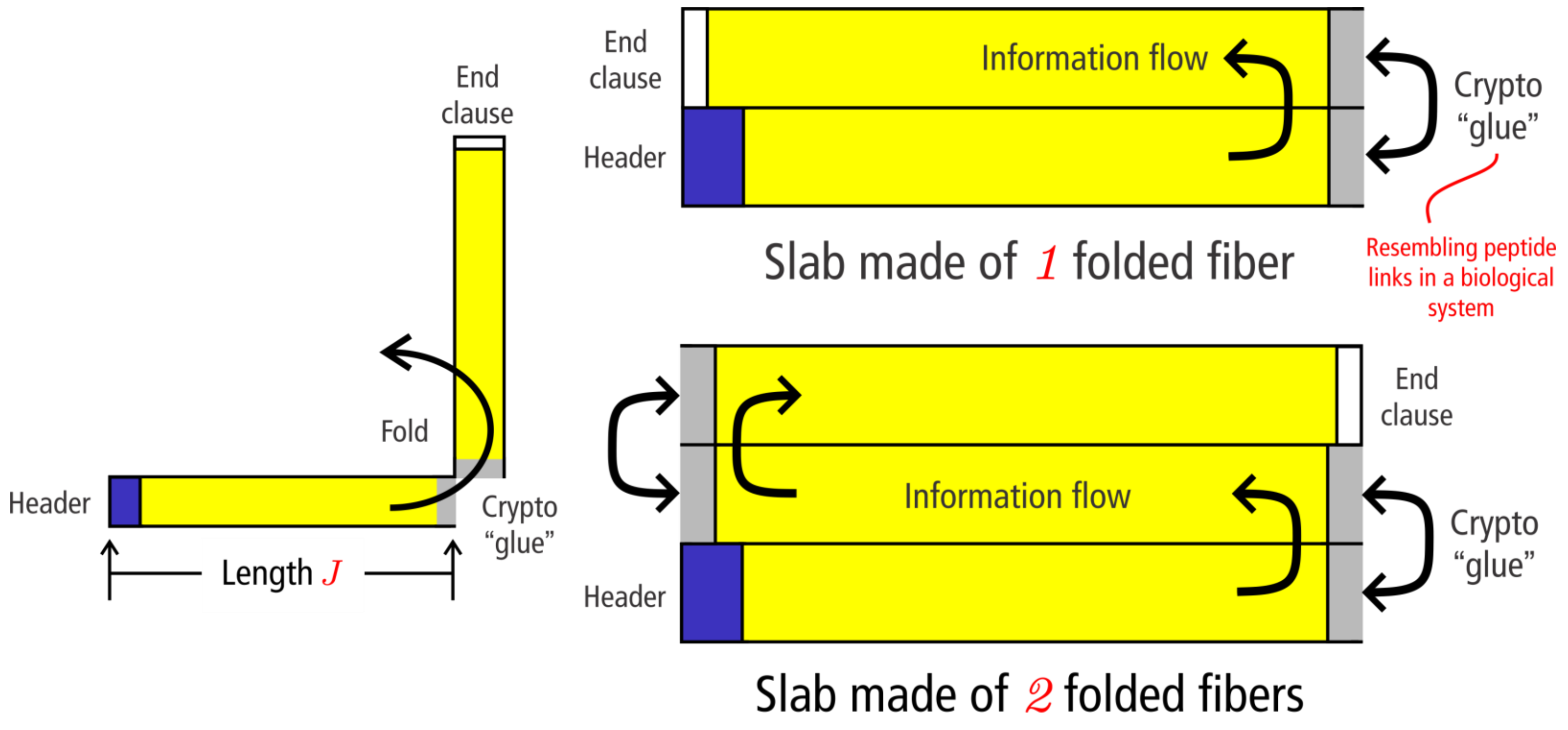


Diagram **1** shows, once again, how K <u>different tokens belonging to the same asset class</u> form a slab.

Row fibers have a length of <u>at least</u> J cells, but there might be instances where the information to be encoded cannot be fully stored into a single row fiber. In **2.a** there are $2J$ cells being used. Cases like **2.b** could also arise, when the number of cells required is not a multiple of J.

Both situations can be easily handled in ARES.

Forctis

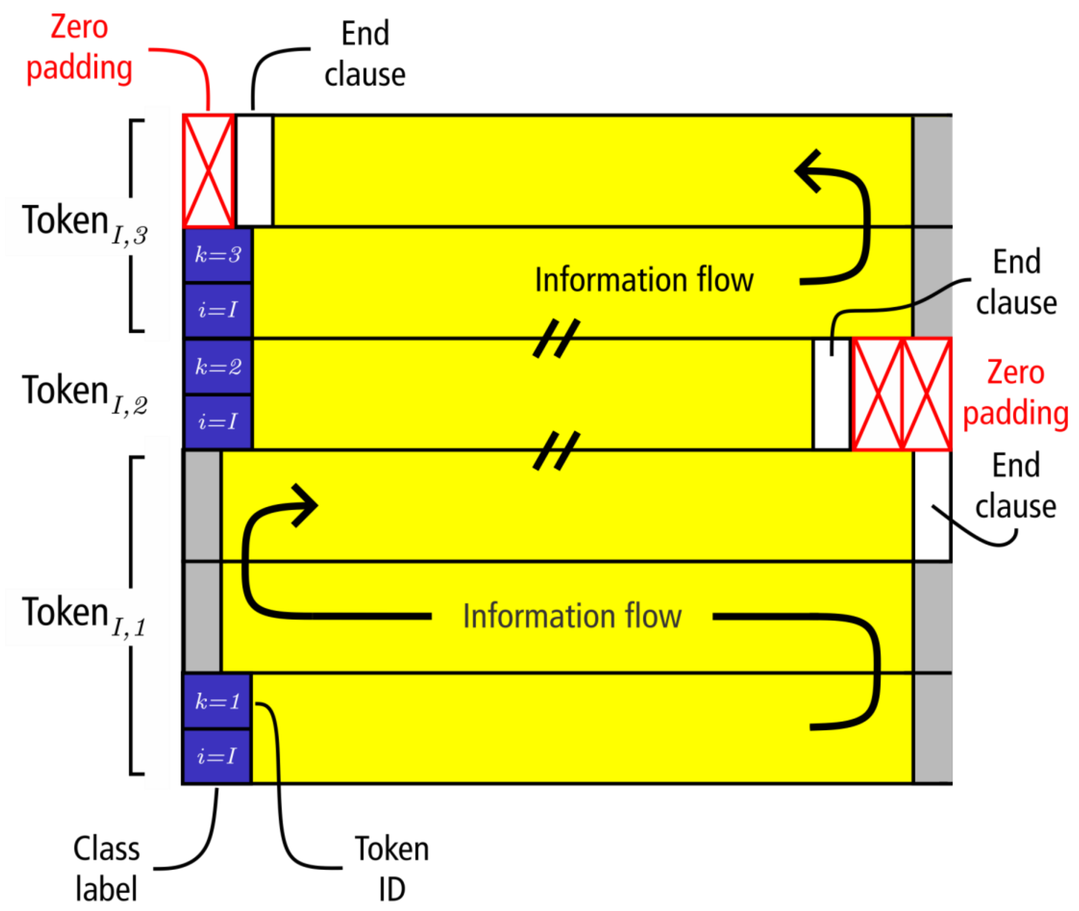


A simple mathematical operation is used to fold an information vector to make it fit into ARES when the encoding of information $> J$ and a cryptographic glue[3] is applied to link the segments belonging individual tokens that are formed by one (or multiple) folding operations. This makes the geometry of ARES quite flexible.

[3] Although algebraically not required, the procedure "hardens" the information encoding and guards against the intentional mixing of vectors. In other words, it ensures each token is indivisible.

Forctis

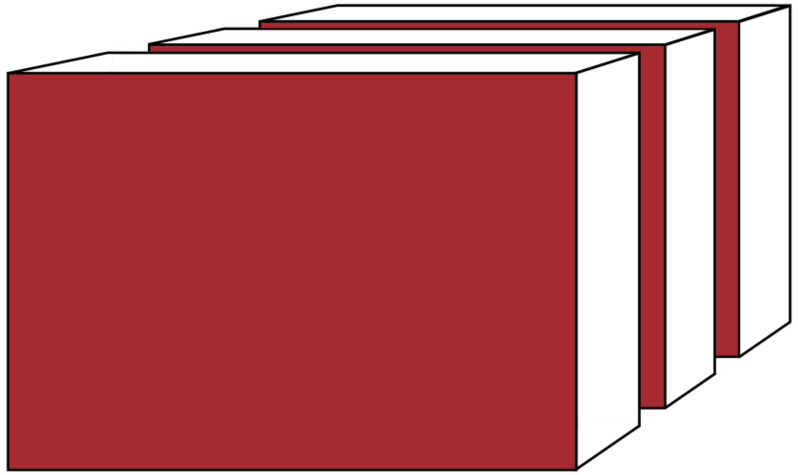
Example showing how tokens for the same asset class (in the same slab) might have a different structure



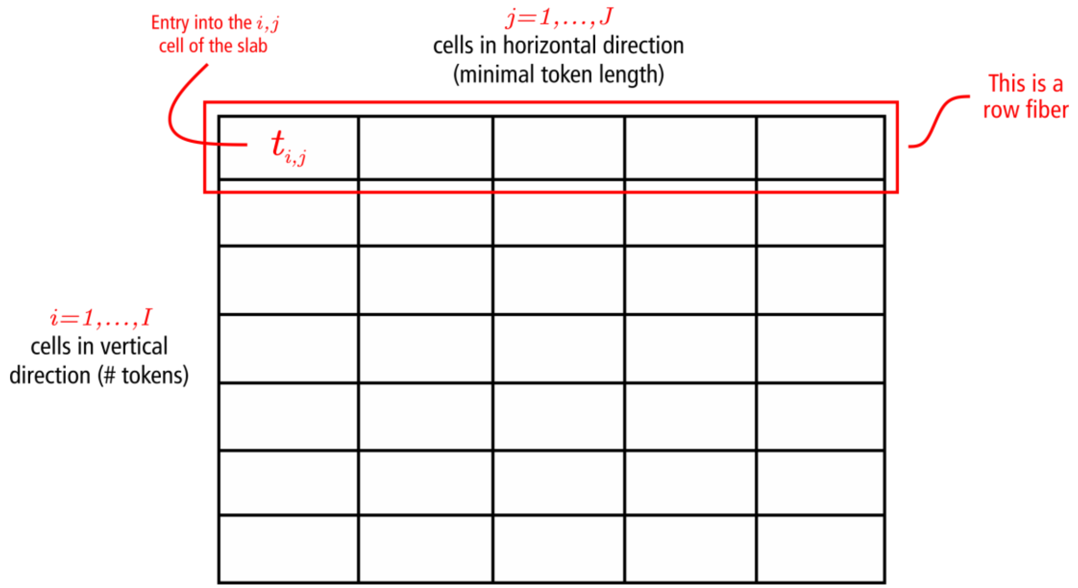
Hence, ARES allows for the combination of tokens, e.g. having a <u>different length within the same slab</u>.

In the diagram, slab I stacks 3 tokens. As before, the differences in the length of each token reflect the number of cells that are required to encode the information stored into them.

- Token 1 in slab I has length equal to $3J$ cells, or 3 folded fibers.
- Token 2 has length J including 2 null cells (zero padding).
- Token 3 has length $2J$ and 1 null cell.

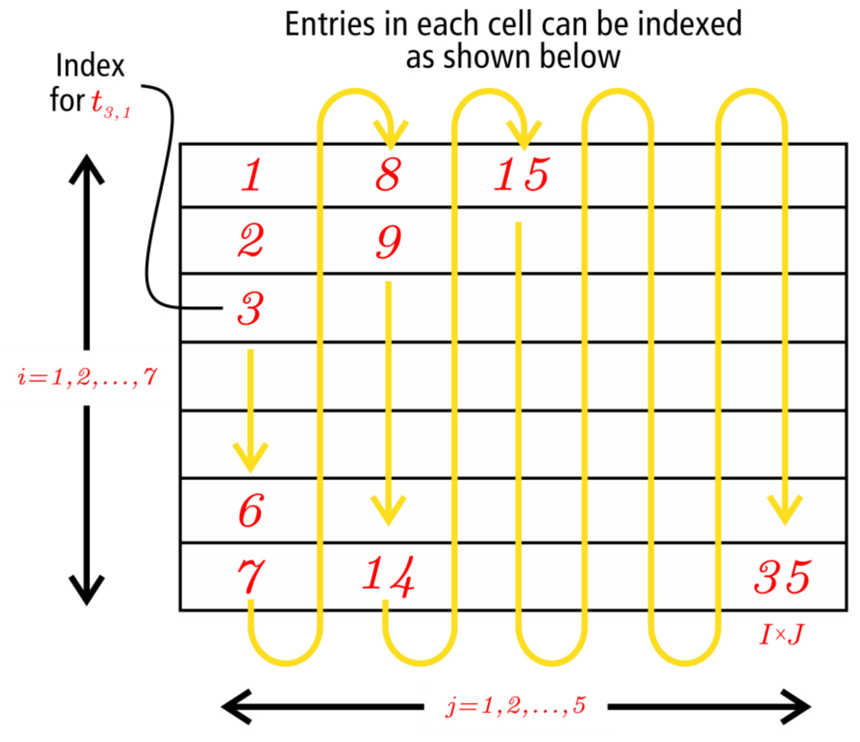


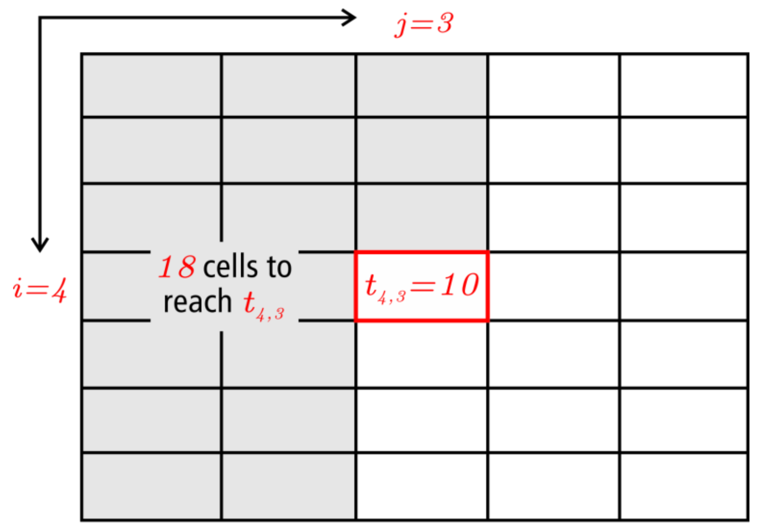
ARES frontal slabs $k=1,\ldots,K$


Entry into the i,j cell of the slab
$j=1,\ldots,J$ cells in horizontal direction (minimal token length)
This is a row fiber
$t_{i,j}$
$i=1,\ldots,I$ cells in vertical direction (# tokens)

Frontal slab data grid

Concerning the partitioning of ARES into data cells perhaps the simplest way to understand how it works is to look at a frontal slab. In other words, a matrix of height I (number of asset classes represented) and width J. If $i=1,\ldots,I$ and $j=1,\ldots,J$ index the rows and columns, respectively, of any given frontal slab $k=1,\ldots,K$ of ARES and the (lowercase) letter t is used to label the cells of matrix T_k then $t_{i,j}$ denotes a cell located in the i^{th} row fiber and the j^{th} column.



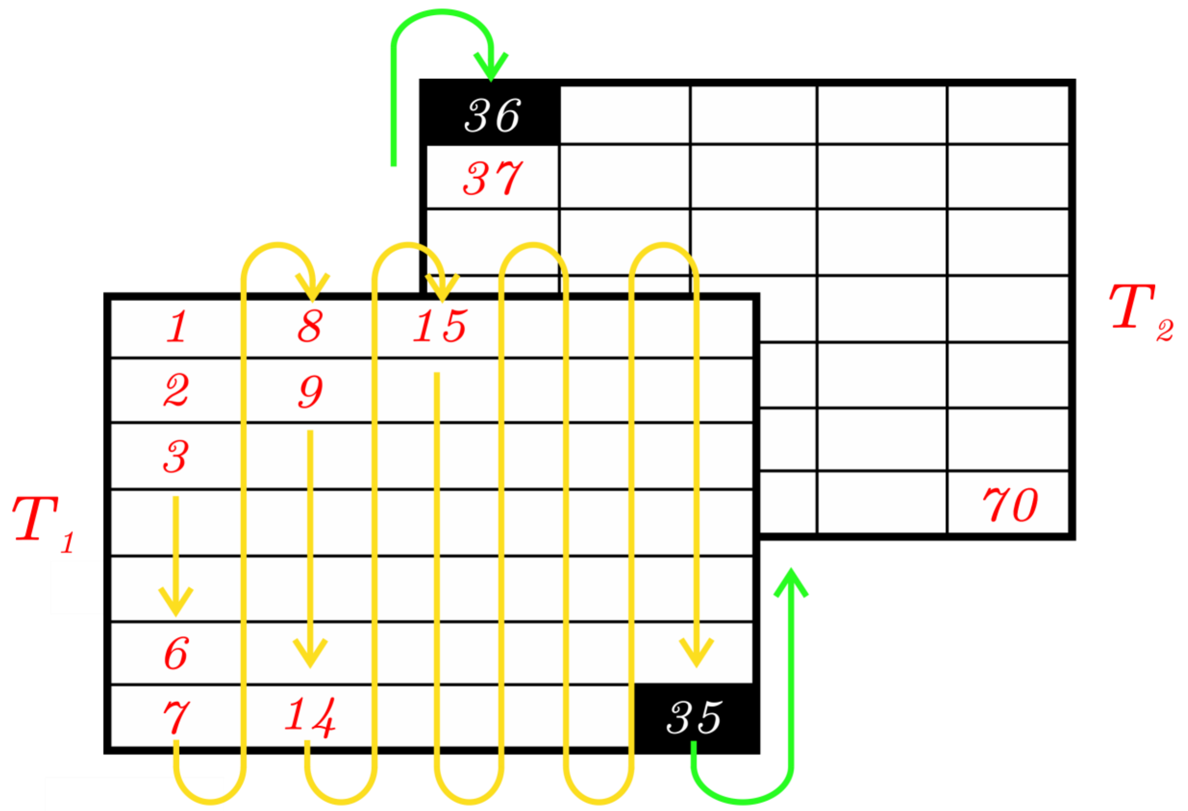


Example: in a slab of dimension $I=7$ and $J=5$ the cell entry $t_{4,3}$ is located in (linear index) position 18. Assuming it stores a value of 10 then $R(18)=10$.

To lift (or extract) the information from ARES a simple way to proceed is by <u>vectorization</u>.[4] This is equivalent to stacking the columns of a data matrix into a single column. If we denote this operation as $vec(T_k)=R_k$ and define a linear index $q=1,..,(I \times J)$ then $R_k(q)$ is the entry in cell q of (frontal) slab k. In the example above, for a slab that has $I=7$ rows and $J=5$ columns the value in cell $t_{4,3}$ corresponding to index $q=18$ is $R_k(18)=10$.

[4] The alternative is to resort to matricization (also known as tensor unfolding or flattening). The interested reader, e.g. can read the seminal paper by Tammy Kolda and Brett Bader, *Tensor decompositions and applications*. SIAM Review, 51(3), September 2009, pp. 455–500. In terms of computation, both methods have their advantages and disadvantages.

Forctis

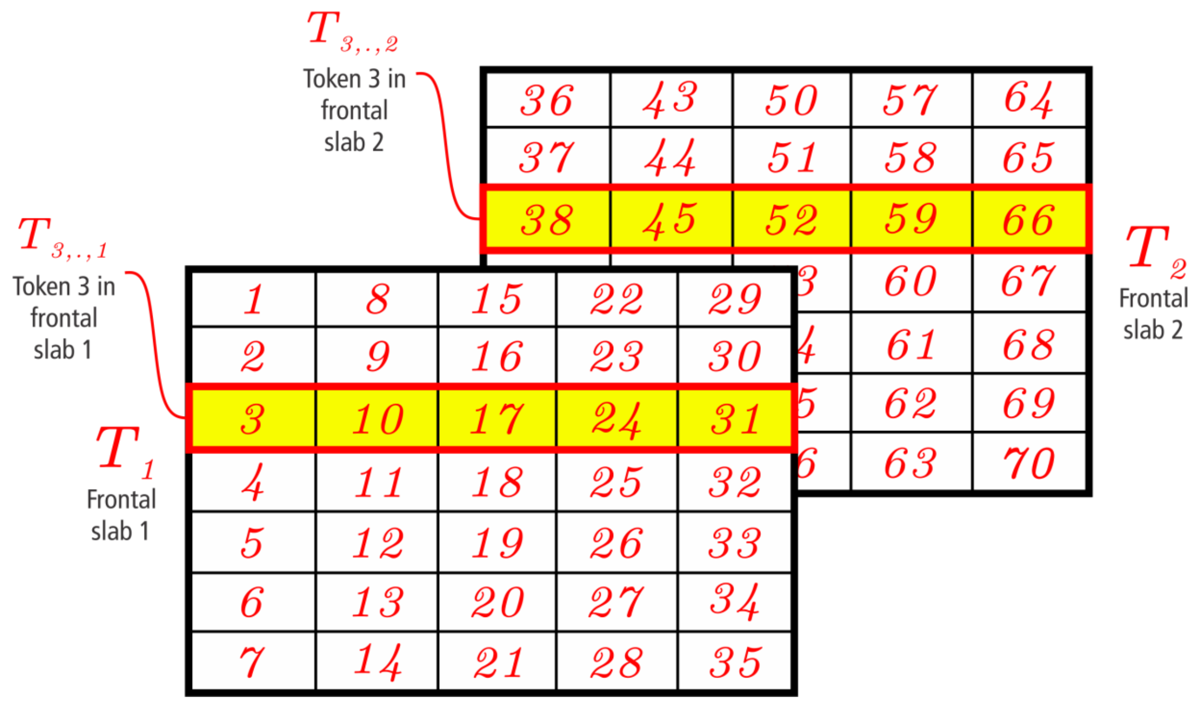


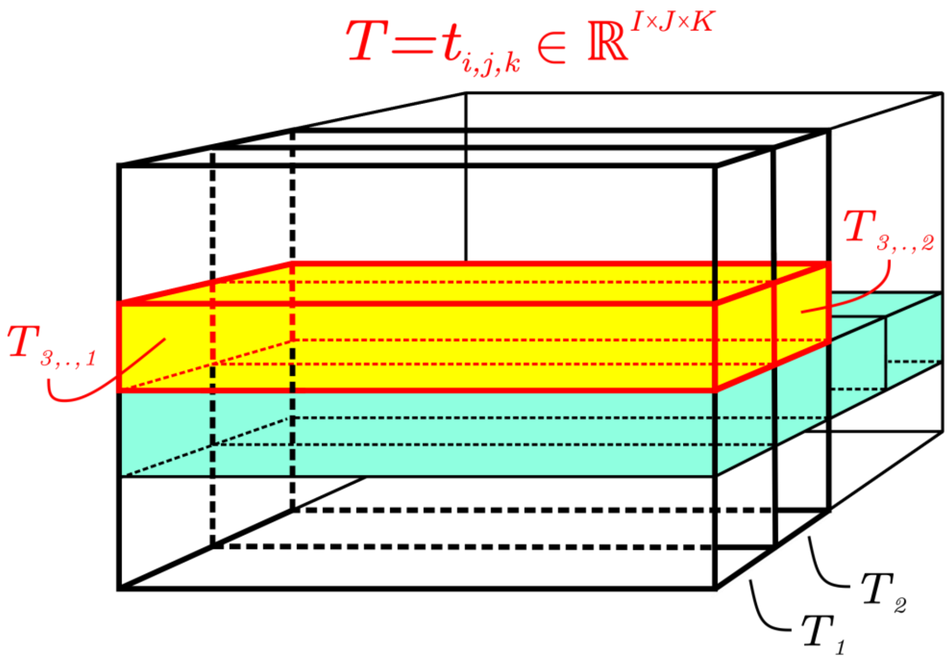
Here we have two frontal slabs (or vertical slices) of ARES showing that the index in position $i=1,\ j=1$ in the second frontal slab follows the indexing sequence that ended in position $i=7,\ j=5$ in the first frontal slab.

The third index $k=1,\ldots,K$ is added to allow the complete identification of any given data cell in ARES.[3]

In this case $t_{7,5,1}=35$ (entry in row 7 column 5 and frontal slab 1) hence $t_{1,1,2}=36.$

[3] In a similar fashion, we can drop the index k to denote ARES as a whole; e.g. $vec(T)=R$ implies that we are not referring to any particular slab of ARES but the whole construct.



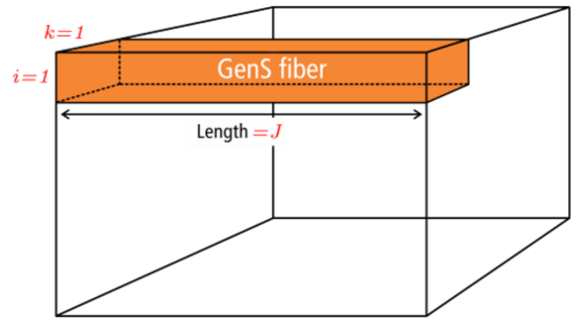


Proceeding in this way, every cell in ARES is indexed. Lifting (extracting) the contents of any token is incredibly easy and very fast. The above example shows the indices for the contents of the token in position 3 whose information spans the cell space contained in two frontal slices.

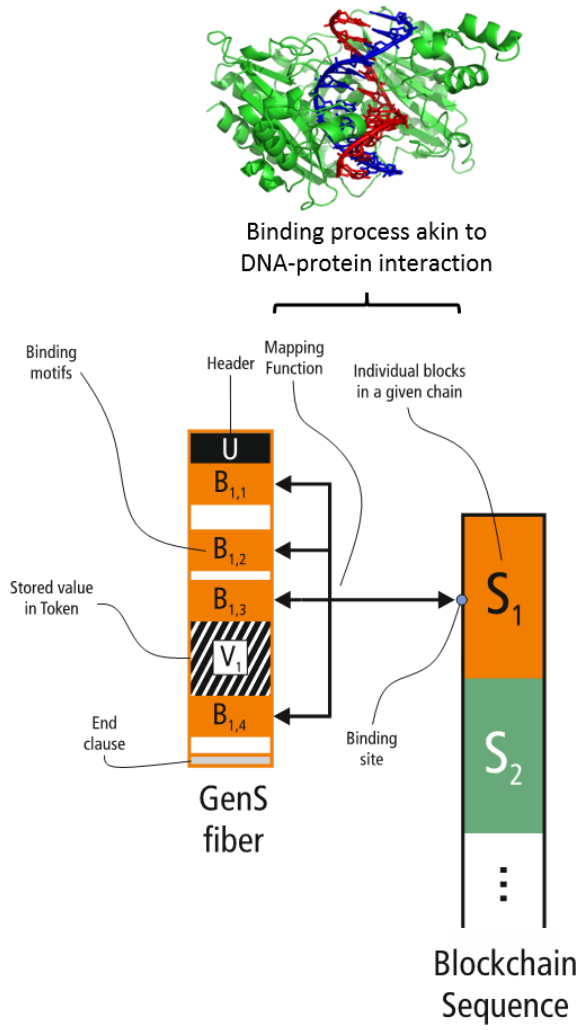
The volumetric representation of this example is depicted above. Lifting requires two pieces of information: firstly, two of the dimensions of ARES, I_{max} and K_{max} (as J is defined by design) that are automatically calculated upon loading; secondly, the desired search position (the $i=1,..,I$ tokens to retrieve) which is a function of the transaction(s) taking place.

ARES as part of our platform

Forctis



The GenS fiber is the essential functioning structure of ARES as it encapsulates the information that allows it to bind to the blockchain. This is a property not shared by any other token stored in ARES.



An essential functional structure of ARES is our stablecoin, going by the name GenS.

GenS is a $1 \times J \times 1$ fiber that is always locked to position $i=1,\ k=1$ and enjoys certain additional properties.

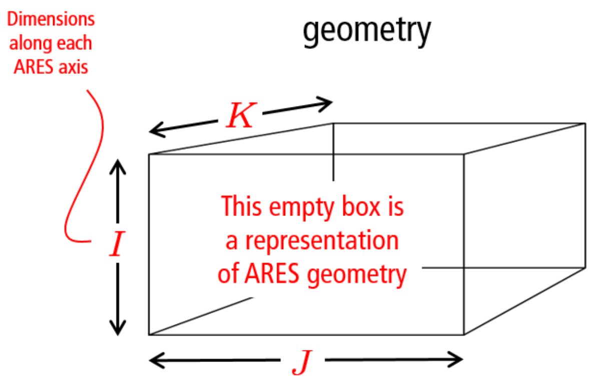
Those properties are the following :

- Its header encodes the UID.
- Its structure includes a cryptographic glue that allows ARES to bind to the blockchain.
- It is a replenishable-type token.

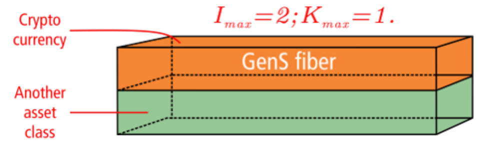
It also de facto makes the top horizontal slab in ARES a cryptocurrency layer.

Forctis

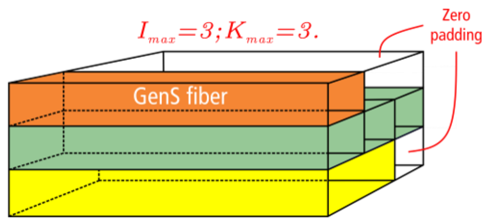
Token geometry


Dimensions along each ARES axis
K
I
This empty box is a representation of ARES geometry
J

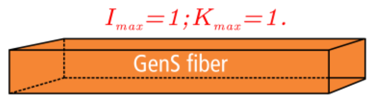
GenS plus a different asset class also encoded as a row fiber.



GenS plus 2 different asset classes having different structure



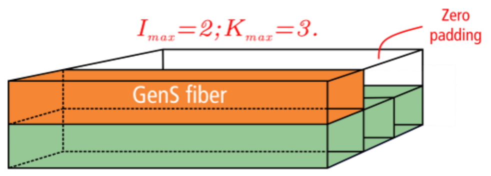
GenS – The essential functioning structure of ARES



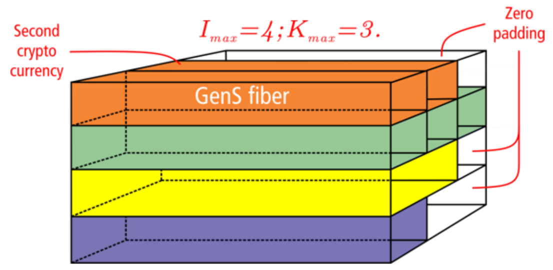
Dimension $1 \times J \times 1$

Design parameter so taken as fixed

GenS plus 3 tokens of a different asset class all encoded as row fibers



Cryptocurrency slab of 2 tokens plus 3 different asset classes with different structure



Using GenS as basic functional block, the geometry of ARES provides a versatile design to encode information in a flexible way and be easily manipulated.

Forctis

The **header** 1 of GenS incorporates three pieces of information: the unique **UID,** a **class label** and **GenS token ID.**

Motifs 2 make ARES recognizeable as a member of the platform by the main node(s), allowing to bind to them. Those motifs are one of the components of the cryptographic protocol that makes attachment possible. There are four motifs per token that are randomly located along their length between the header and the end clause. Consequently, every token has the same structure but not the same layout.

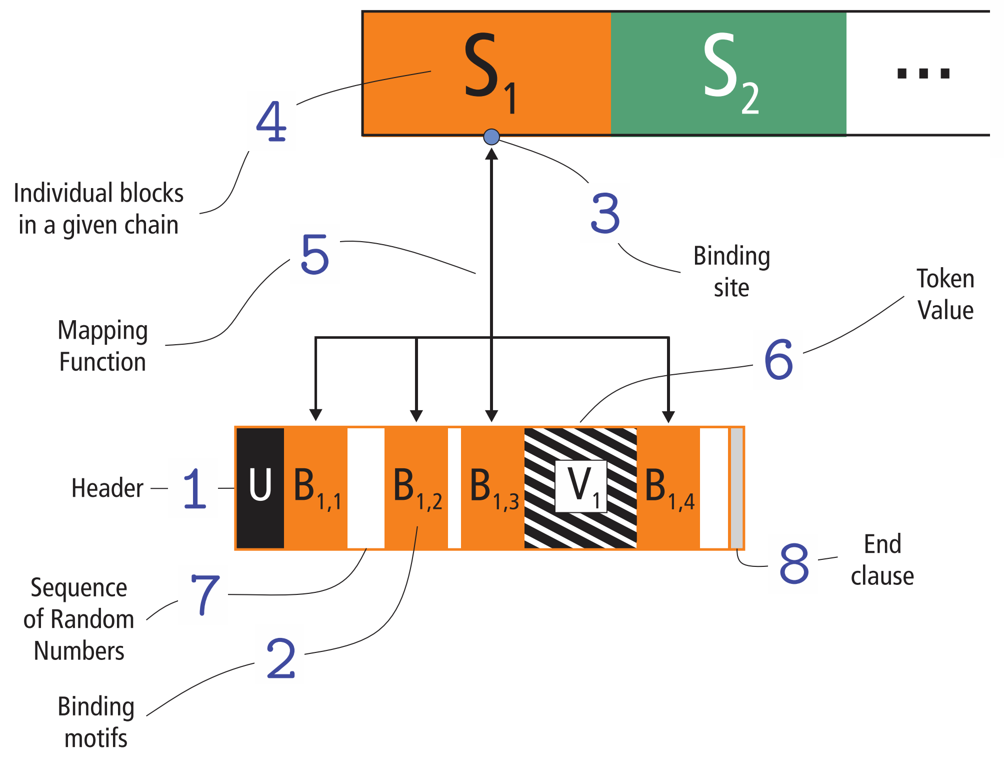
Each master node in the platform hosts a strand of **binding sites** 3 to which every ARES token attaches. There are also four binding sites uniquely associated to each GenS token.

The binding sites are also cryptographically linked together 4 to form a **strand**.

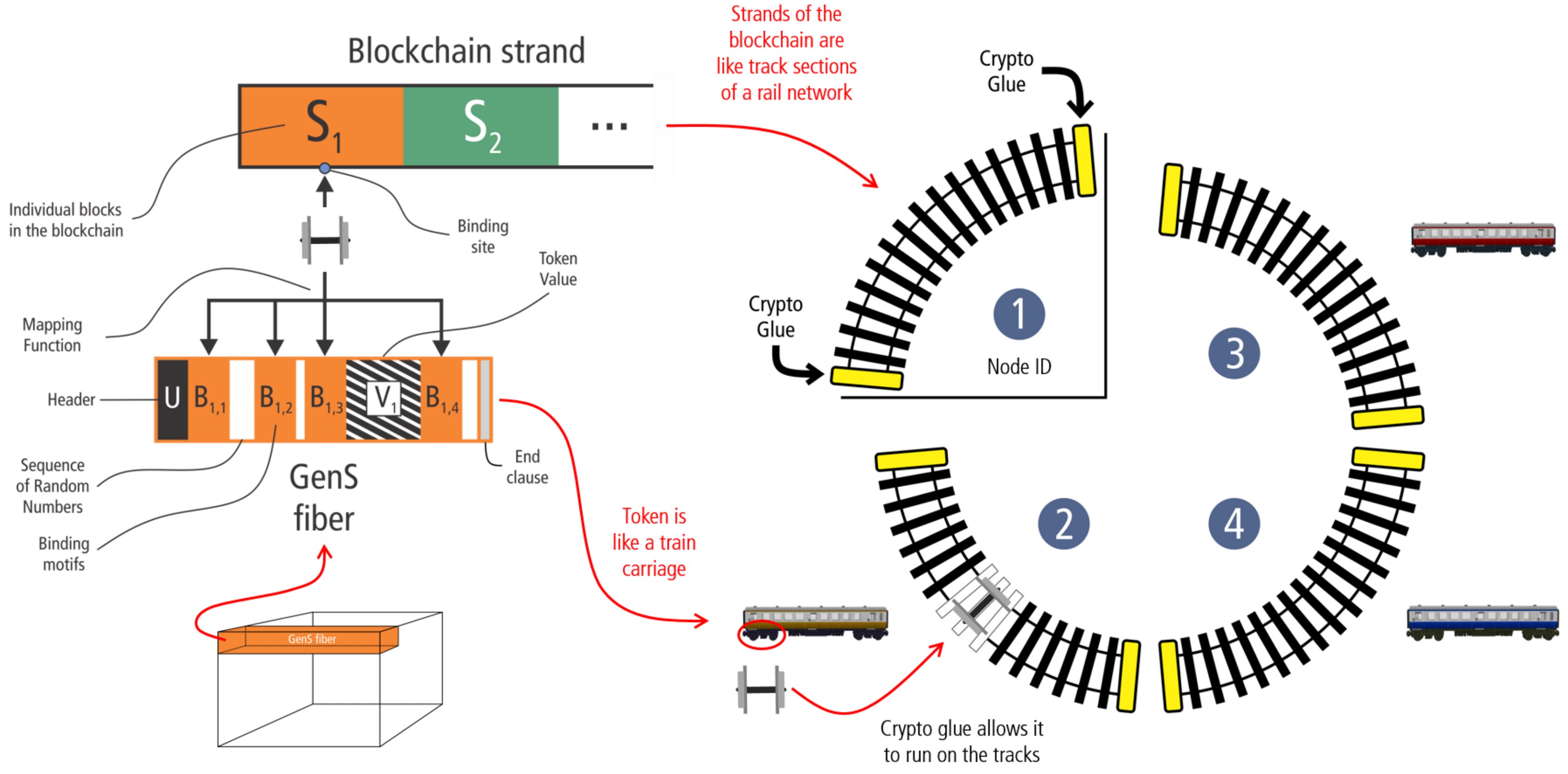
A **mapping function** 5 links the token motifs to the chain via the binding sites. This is another component of the cryptographic protocol. Recognition needs two or three motifs (randomly chosen) to pair to the binding sites. Each time an ARES token seeks to join the platform, those motifs are picked from the pool of four available (and unique to each token).

A specific section of the GenS code 6 is used to **store its value.** This is effectively a counter that gets updated on every transaction and can, of course, go to zero.

There are additional cells 7 purposedly placed in each token (as a masking device) to be filled-up with **random data** and a specific section at the end of the token 8 is reserved for the data designating **where a token ends**.



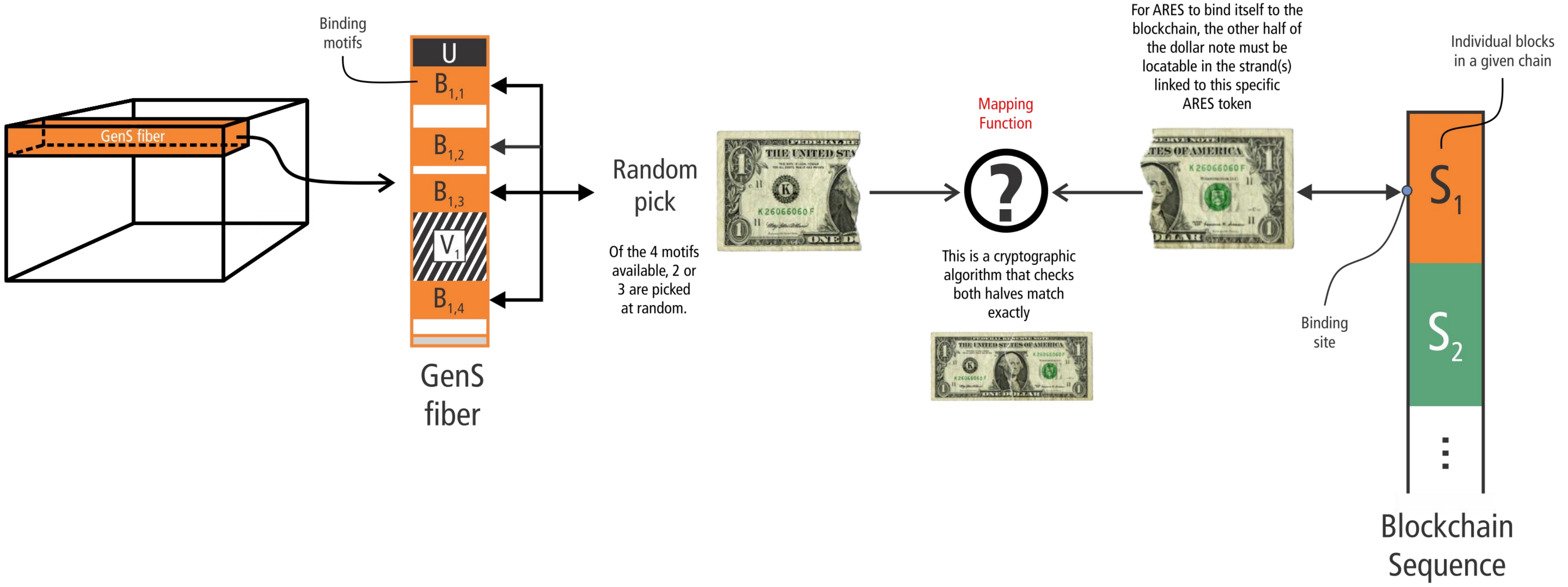
Forctis



Blockchain strand
S_1
S_2
...
Individual blocks in the blockchain
Binding site
Token Value
Mapping Function
Header
U
$B_{1,1}$
$B_{1,2}$
$B_{1,3}$
V_1
$B_{1,4}$
End clause
Sequence of Random Numbers
Binding motifs
GenS fiber
GenS fiber
Strands of the blockchain are like track sections of a rail network
Crypto Glue
Crypto Glue
1
Node ID
2
3
4
Token is like a train carriage
Crypto glue allows it to run on the tracks

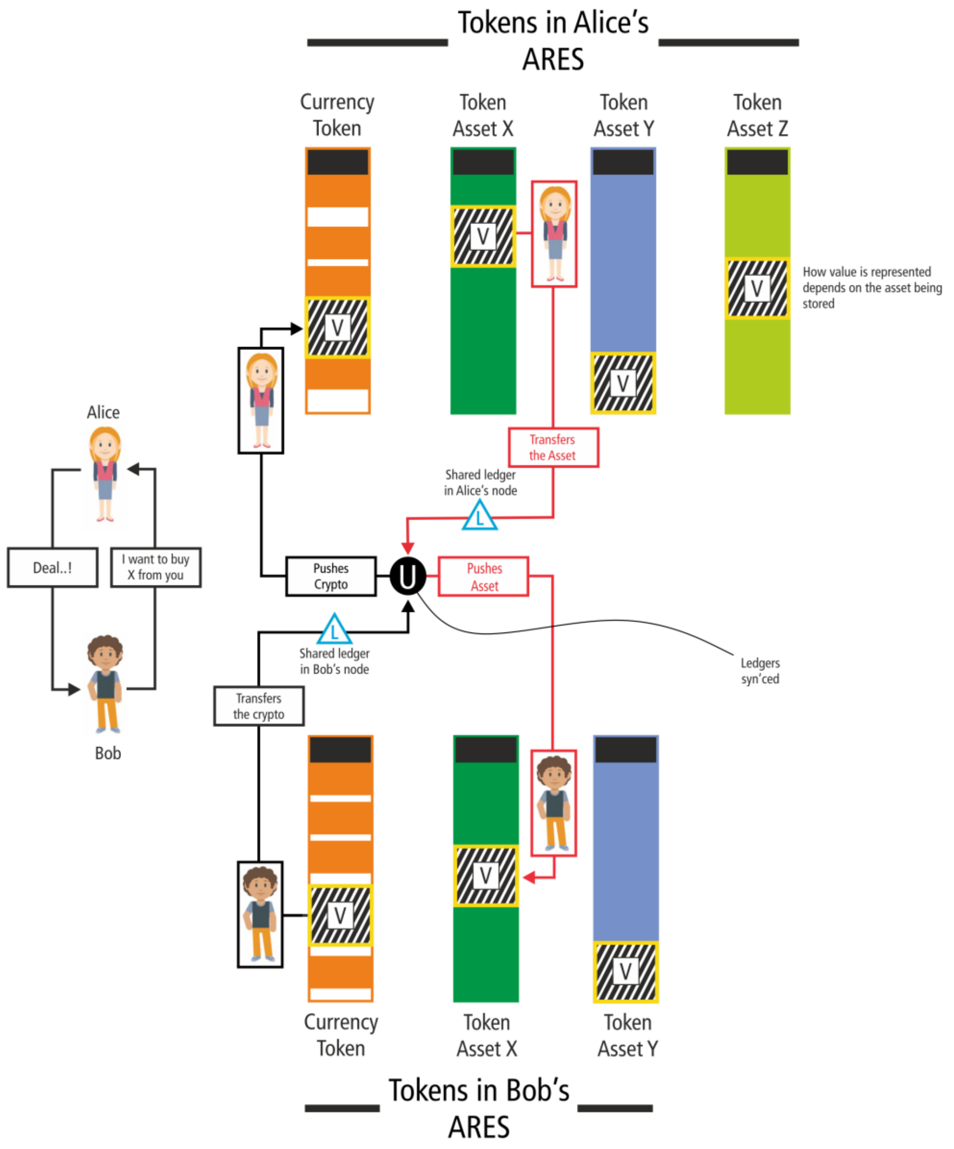
Each ARES token is like a train carriage that runs over the tracks that make the blockchain. Each section of track (or blockchain strands) is independently managed at the node level. For the carriages (our tokens) to be able to run on top (or equivalently, bind to) and be able to travel along the tracks (in order to exchange information) they must have the same gauge as the track. That gauge is one of the cryptographic glues in the protocol, anchoring token motifs to a particular binding site along each chain.

Forctis



GenS fiber
Binding motifs
U
$B_{1,1}$
$B_{1,2}$
$B_{1,3}$
V_1
$B_{1,4}$
GenS fiber
Random pick
Of the 4 motifs available, 2 or 3 are picked at random.
Mapping Function
?
This is a cryptographic algorithm that checks both halves match exactly
For ARES to bind itself to the blockchain, the other half of the dollar note must be locatable in the strand(s) linked to this specific ARES token
Individual blocks in a given chain
S_1
S_2
⋮
Binding site
Blockchain Sequence

A simulated representation of how ARES binds itself to a node in the blockchain using the information coded in GenS. The coat of crypto-glue applied to GenS allow ARES to stick to a participating node in the network and thus be able to interact with other ARES tokens.

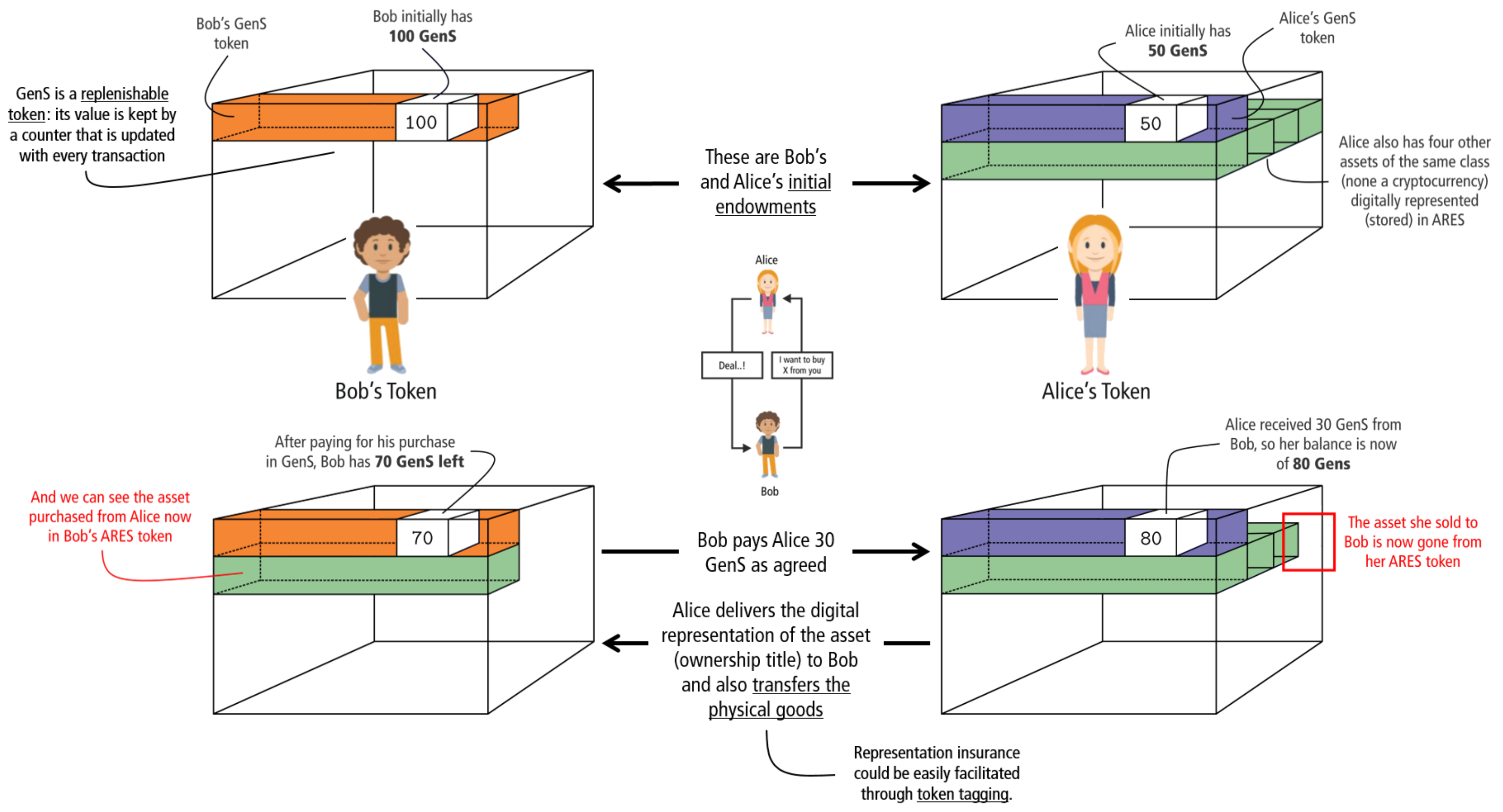


Transactions amongst holders of ARES tokens involving different asset classes can take place <u>consistently and efficiently</u>.

Smart Contracts could be used to regulate such events, when necessary.

Even more, going beyond Smart Contracts the protocol opens the route for the algorithmification of transactions.

Forctis



Bob's GenS token
Bob initially has 100 GenS
GenS is a replenishable token: its value is kept by a counter that is updated with every transaction
100
Bob's Token
These are Bob's and Alice's initial endowments
Alice initially has 50 GenS
Alice's GenS token
50
Alice also has four other assets of the same class (none a cryptocurrency) digitally represented (stored) in ARES
Alice's Token
Alice
Deal..!
I want to buy X from you
Bob
After paying for his purchase in GenS, Bob has 70 GenS left
And we can see the asset purchased from Alice now in Bob's ARES token
70
Bob pays Alice 30 GenS as agreed
Alice received 30 GenS from Bob, so her balance is now of 80 Gens
80
The asset she sold to Bob is now gone from her ARES token
Alice delivers the digital representation of the asset (ownership title) to Bob and also transfers the physical goods
Representation insurance could be easily facilitated through token tagging.

Say Bob wants to buy an asset owned by Alice, using the cryptocurrency GenS, of which he has 100 units. Alice agrees and they convene a price of 30 GenS.

Forctis

This deck outlines some of the basic principles guiding the conceptual design and engineering of **ARES**. It is meant to provide the reader with a basic understanding of what ARES is, how it is build (as a mathematical object), what are its possibilities for storing digitalized assets and present the role that **GenS**, our (stable) cryptocurrency, plays in it.

The presentation is by no means exhaustive, so please contact us to info@forctis.io if you have any additional queries.

Forctis

http://forctis.io

This document and the information contained therein are © 2017, 2018 of Forctis AG. We kindly ask you not to distribute this presentation or parts thereof by any means but to instead direct interested parties to http://forctis.io/resources for the most updated version.

For any queries, please contact us at info@forctis.io.

Logos and isotypes are ® by Forctis AG. Images used are © of their respective authors.

Forctis AG is a company registered in Switzerland under number CHE-410.610.981
Registered address is Samstagernstrasse 55, 8832 Wollerau, Switzerland

Version 3.1 September 2018