3/28/2018

GONT White Paper (extended version)

Contents

Introduction

GONT solves on of the Blockchain's fundamental problems: nowadays, tokens are used as an instrument for speculations, but no one uses the "food" tokens.
I.e. there are no truly practical "service" tokens on the market.

Nowadays, there are no truly useful mass services (except of stock exchanges and suchlike) which would work using tokens.

In order to give Blockchain a real (non-speculative) value the tokens of the new generation are needed (**Token 3.0**).
To that effect GONT introduces the concept of the service gas (or AlGas – the alternative gas). We can talk about the "service gas" as about the "alternative gas" as it's the orderly development of the gas idea into Ethereum – in the same way that Alt-coins develop the Bitcoin's ideas.

For protection of the already-existing investments into development GONT always keeps the full backwards compatibility with Ethereum, but upgrades some certain components.

1.
What's the essence of the GONT?

GONT is the Blockchain platform for mass services, which are likely also comprised of the plenty of **macro-services** performing "real" or "intellectual" work. And that's the strong basis for introduction of the new strong crypto-currency.

GONT introduces mechanics for recording the work of all micro-services by means of the technology of virtual machine's service gas. Hence, **GONT takes the "monopoly" for the introduction and development of gas economics from Ethereum**.

This strengthens the decentralization of the full system, making services the economic subjects instead of objects.
Services participate in assessing gas' cost and its technological complexity on their own.

1.1 Sharding at GONT

While some projects create sharding on the basis of the "blockchain of the blockchains" [TON, Plasma, etc] conception, GONT doesn't rejects this approach but interprets sharding more extensively.

Sharding at GONT shall solve not only the problem of the Blockchain scaling and significant productivity rate but also become the basis for the coherent and reasonable crypto-economics based on the service gas' physical work.

That's why GONT-sharding solves two global tasks at the same time:

- Blockchain scaling
- Introduction of the service gas

GONT also gives its advanced interpretation for:

- transactions at Blockchain
- STATE (both State EVM, and State transactions)
- **EVM – implementation through GONT VM and gVM itself**
- smart-contract
- consensus
- functional space of possible gas work (for services)

GONT also strengthens the concepts of the Blockchain "self-treatment" and self-verification

1.2 GONT-BUGIS (Bug-IS) and GONT-BIST (Built-in-self-test)

Anti-bug GONT system (or the Bug-IS system for forecasting bugs)

BUGIS and **BIST** is a system for the formal verification, integrity regulation and protection from development faults.

It's also a system for the automatic code generation from the meta-descriptions of services in the GONT-Space.

From the viewpoint of the user GONT implements the **one-button-click** (Apple style) interface, i.e. the conception of the automatic service generation using simple meta-descriptions (in the GONT-Space tree). Furthermore, the service will guaranteed correctly work with GONT and Blockchain.

The user doesn't need to explore lots of implementation details. It's enough to press one button for getting the minimum working service (**one-button-click**).

This allows introducing the basic "**Golden-model**" for the gVM core and further systematic expand of it by the code generation from meta-descriptions in the GONT-Space.

Hereinafter the code will also be automatically generated for the HW core version (in the ASIP generation style) as well.
GONT makes reincarnation and significantly strengthens the ASIP market.

2. FUPE Space

2.1 FUPE Space as a hybrid SW/HW space

GONT has the GONT VM (gVM) generation technology, which is the EVM analogue, based on the meta-information about the service.
I.e. the each new service within GONT affects the "EVM" structure. "EVM" is not a static entity (unlike Ethereum EVM).

FUPE пространство

FUPE = FU (Functional Unit) + PE (Processing Ellement)

GONT offers to transfer from the conception of "simple gas – simple transactions" to the conception of "complex gas – complex transactions". The service gas' complexity is determined by the services' creator and it may be selectable.

I.e. GONT interprets the gas more complex.

FUPE and quarks, IoTT

Services in FUPE can be implemented through the vast number of various quarks. These quarks can cover the considerable amount of the economic activity fields. From the management of petrol stations and cashier tills in stores to streaming video encoding/decoding. We want to include all types of hardware accelerators into the GONT Space. And any and all possible processor architectures (ARM, MIPS, ARC, VLIW, etc.).

It can be said, that FUPE Space will be IoT-full.

2.2 FUPE Space and GONT – for the coherent growth of the decentralized economics

About how the FUPE Space and the AlGas based GONT changes the common economic model on the basis of money.

Regulation and exits to the "Economic freedom"

The new economic reality of the crypto-currency has faced with the regulation and the "**man-in-the-middle attack**". I.e. after all, the possible middlemen-regulators have been integrated into the so-called decentralized system of crypto-economics. That will lead to the "edge effects" of all kinds and the significant capitalization decrease of the crypto-world in the very near future.

To solve the problem it's important to offer the market legal mechanisms for "anti-control" of the entire system. It's obviously, that the regulation will kill all the crypto-economics' capitalization after all, but the well-circumspect "anti-control" (not only against the middleman-state, but also against the centralization of the "power" in the hands of the very main blockchain projects) will ride out the situation to the normal development direction.

FUPE-based GONT offers both new anti-regulatory mechanisms (economics, which is impossible to control) and mechanisms for creating the new economic valuables.

The "man-in-the-middle attack" is an attempt to take under control any valuables exchanges between subjects and force new exchange rules.

The Ethereum-based Blockchain solves this problem less than fully for several reasons.

- The "man-in-the-middle attack" was a success for the currency convertation (fiduciary crypto-currency). I.e. all the stock exchanges and exchangers are already controlled via KYC.

- The "man-in-the-middle attack" is initially implemented to the system's "gas" by the Ethereum creators themselves, and there are no signals for changing situation. I.e. independent subjects don't have any scenario for creating "their own" service gas, which will conduct the real transactions work. **From this point of view it's still necessary to conduct the decentralization of the Ethereum gas and integrate the alternative gas – AlGas.**

Valuables and GONT

GONT works with a variety of valuables:
- Fiduciary currencies
- Crypto-currency and tokens
- Services

The linker between all the valuables' types is AlGas
AlGas is the main GONT philosophy, which also allows talking about the considerable GONT innovation in reference to the competitive systems.

The concept of the structured ontologies

Injecting new knowledge into GONT Tree is conducted according to the structural rules. In fact, according to the rules of the language, what won't allow breaking the structural integrity of the GONT Tree.

Each user from the lawyer to the script writer will be able to fill the assigned cell of the GONT Space in the simple way.

For the mass market capture it's necessary to operate the ontologies' **macro-units**.
Otherwise all the processes will be stuck in the routine works.
For working with the GONT Tree another optimum alternative is to introduce the certain macro-language.

The plenty of the units' suppliers for FUPE

There's a great plenty of the manufacturers of hybrid (Tensilica, etc) and ordinary (ARM, etc) processor architectures in the world.
There are also lots of consortia exploring this problem (HiPEAC, etc).

All of them can become IP providers for FUPE.
Through the mechanism of a special FUPE-Gas they will be able to receive roaylty.

Characteristic properties of the AlGas customized currency

- Depositary essence
- Advantages over the "centralized" Ethereum tokens

SWOT (part 2.2)
- Can a third person begin to control gas flow in order to try to conduct a "man-in-the-middle attack" and become a middleman-regulator (the most obvious thing is the state wants to do it)?
This will violate the balance of private ownership for the gVM virtual machines.

Characteristic properties of the economics around FUPE
- A great variety of the mutual offset transactions (hybrid gas and suchlike) is vastly larger than in the ordinary economics, when the number of the service providers is small. FUPE allows introducing lots of thousands providers for the service.

- Services without the owner

Services without the owner (the collective AlGas consumption) exist as in the ordinary economics, but can’t become a target for the "man-in-the-middle attack".

Economics types for FUPE

FUPE is relevant both for the pure digital (computational) economics and for the real one (for example, autonomous physical IoT objects, Blockchain-cash desks, etc.)

For example, FUPE can fully implement the customers projecting laser shows. The communication with the real world is conducted only through the electricity consumption. All the additional cost remains in the GONT system.

FUPE and launching traditional mobile applications

The Dalvik virtual register machine at all types will also be the part of the FUPE Space construction units. The work of the Davlik will also be regarded within the AlGas conception.

And this means, that the user will have an opportunity to launch traditional mobile applications, for example at the blockchain-mobile phones, blockchain-cash desks and other devices.

Local conclusions (part 2.2)

- GONT and FUPE allow removing from under the "man-in-the-middle attack" a lot of economic segments and return the world to real economic freedom, to the world without middlemen.

- According to the above, our aim is to settle the maximum number of partnerships with industrial players (Microsoft, ARM, Tensilica,..) for expanding the FUPE Space. At the same time, FUPE Space will be expanded by processor units of its own development (or mapping from the market developments).

2.3 FUPE Fundamentals

The basic reason for introducing FUPE is the need to limit smart-contracts according to the "necessary and sufficient" functionality for complying with various trade-offs – compromises of the integrity and efficiency of the entire system.
FUPE will be closed-loop to the specialized ***GOL functional programming language***.
Fundamental units for FUPE will be selected very carefully.

GONT VM
Virtual machine of the new generation!

The traditional Ethereum EVM-based model of contracts needs ideological expansion by transferring to the more complex dApps (beyond the Solidity conception).

For example, for launching Android applications Dalvik VM is used, which is a register machine. Whereas, EVM is a stack machine (for simplification and security – as it has been expected).

That's what, if we want to create something like a blockchain-phone [Sirin Labs], we need to come over to the register machine.
But it'll be even better if the new VM can support both the stack part (EVM) and the register one (Dalvik) by clustering the functionality.

We've created such the VM – **GONT VM**.
It's the fundamental container for the entire FUPE functionality.

At the same time, the GONT VM has to be (and is) the "cyclic equivalent" for rtl (Verilog, VHDL) code – for further implementation within HW.

Hybrid languages of the contracts

There are a lot of languages (in addition to Solidity) for writing contracts and some of them are successful.
For example, the NEO project supports contracts at .NET and has settled the partnership with Microsoft [], which is good for the project's capitalization.

AlGas and PIPE of any length
During the transactions conducting the transaction channel (TREVAL) is created that implements a functional Pipeline of the random length

Pre-Exec and routing

Pre-Exec Conception gives a method for implementing the dynamic creation of the channel for conducting transactions.

FUPE classification

- Simple: Registers, ALU, FPU
- Complex: VES

Data types

GONT expands possibilities of the data stream processing.
gVM cores have an interface for working with large data streams, for example video.
For example, this option will be relevant for the video encoding gas.

Requirements for the gVM core for implementing the AlGas approach

- Automatic assemblage from the meta-information (about instructions and registers)
- Availability of both cycle accurate and fast model (the reason, why QEMU is not suitable is absence of the cycle accurate)
- Availability of the interface for verification in reference to the rtl code (for further implementation within HW).

Examples of the GONT ONTO "assets" (GONT assets)

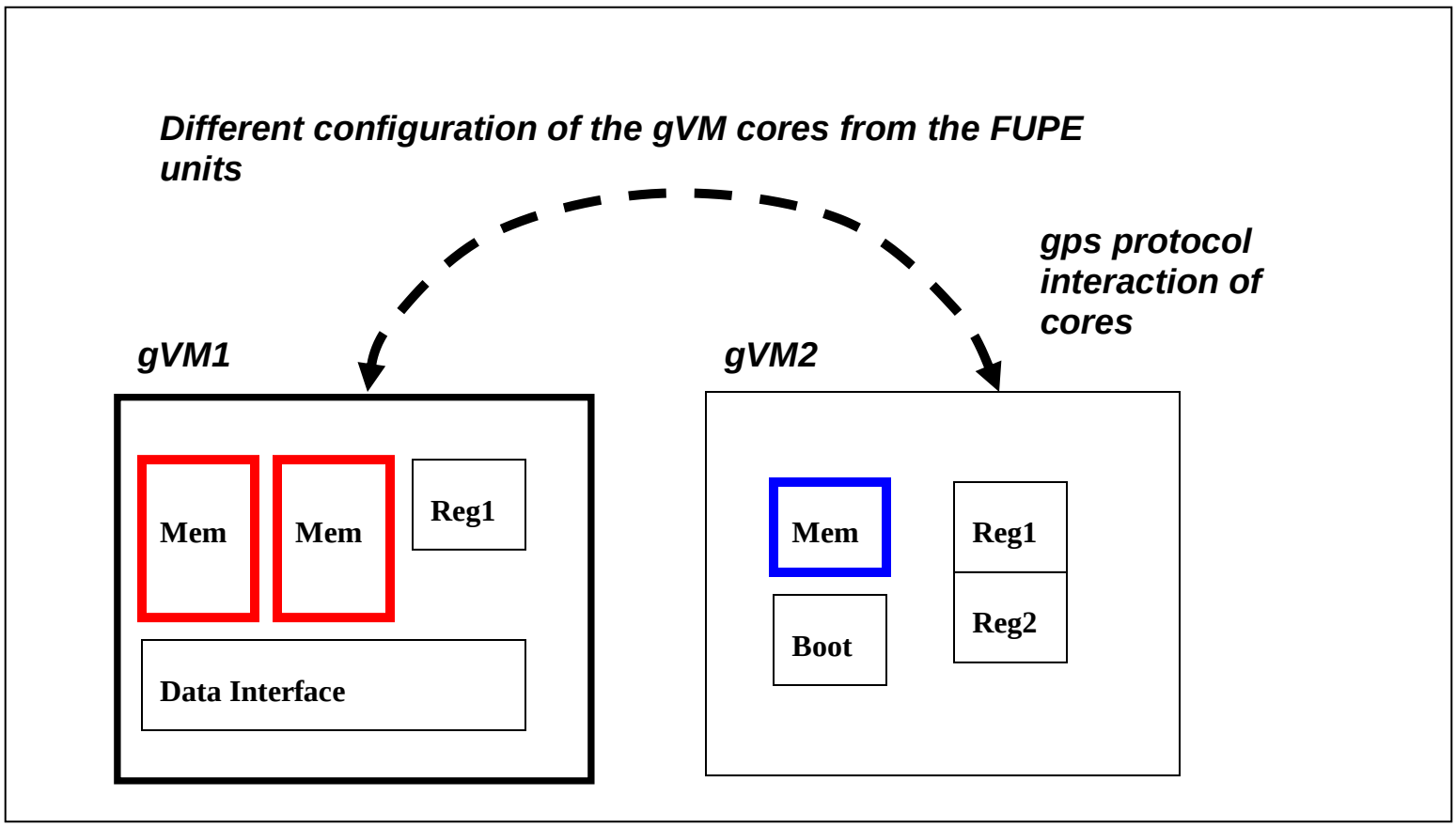
- Morozevich Line in Chess
Motivation: Promoting one's virtual assets into the history
[https://www.youtube.com/watch?v=I6h9r4xgPFM]

- Klyshko scattering in physics (named after the ground breaker), and not just spontaneous parametric scattering.
Motivation: Securing the personalized "asset" to the discoverer

2.4 FUPE units for EVM

FUPE gives an opportunity to implement the dynamic constructor method of cores with various parameters (proper for the AlGas tasks)

For example, one gas requires the machine with the memory of **1 Kb**, and the other one – of 1 **Mb**. Or when the machines of the different productivity rate are required.



gVM machines generated for different FUPE parameters

In such a way it was concluded, that the machine generation process itself becomes the Al Gas function. I.e. the generation of the different machines itself is interpreted according to the different prime cost.

Let's introduce the concept of the FUPE-Gas
- Generation of the gVM machines from the FUPE units, which is recognized through KANT.fupe gas. This gas is used for payments to the creators of the FUPE units for the gVM machines.

Which dynamic parameters of the machines and the FUPE units may be considered?

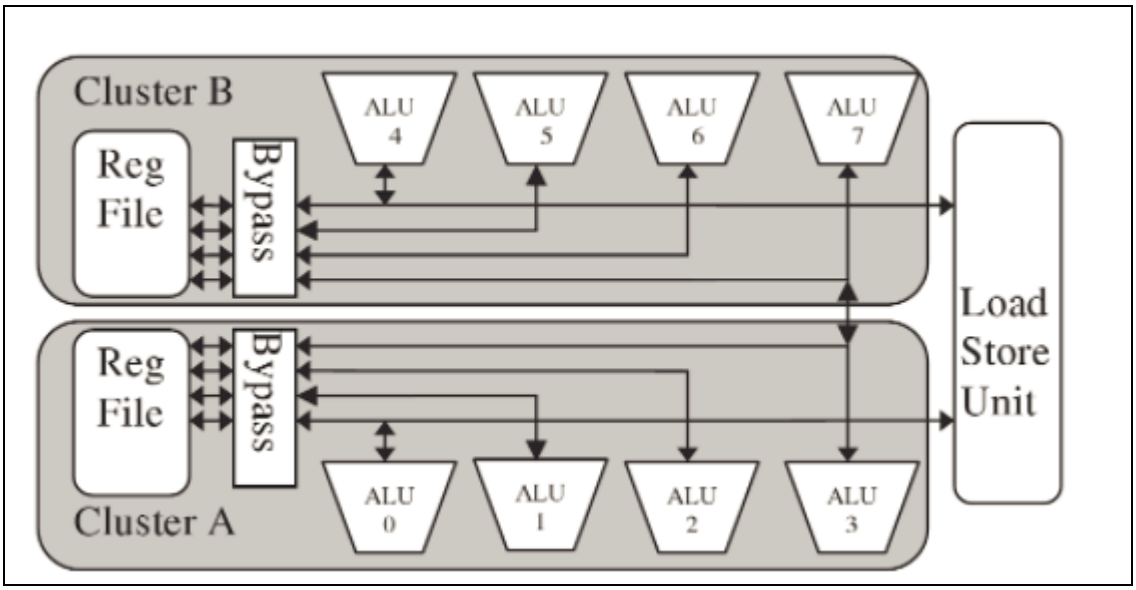
For example:

- Main Memory Size (Mem)
- Boot Size

- Availability of the interface for working with data (data streams)
- Trace availability
- Amount of the register files and their volume
- TLB size and amount
- Availability of the operating system (+ type selection)

The multi-clustered EVM

GONT uses the multi-clustered EVM approach for the logical separation of the business-units. For example, for the simultaneous support of one EVM of binary codes from various blockchains.



Example of the multi-cluster

Herewith the “old” EVM Ethereum is equivalently displayed to one of the clusters. I.e. GONT VM expands EVM by means of the cluster approach.

Conclusion (part 2.4):
We set the parametric assemblage of cores from the FUPE Space elements.

2.5 Additional properties of GONT transactions

1) **TR can be "expanded" through time**

Until the moment of confirmation, TR can last (conventionally) both a year and a millisecond – depending on the TR meaning.

2) TR can consist of other (included) TR

3) TR can be implemented at the service gas from different owners-services

4) TR can be implemented at a large variety of quarks (ARM, VLIW, etc)

5) TR can have the communication with the real world by means of the IoT devices

6) TR exists so far as the meanings describing it at the GONT Tree oracle exist
TR is also implemented through many contracts

7) TR cost consists of the prime price and owner's margin (profit) (through the AlGas system). The cost of the transaction includes the miner's margin as well.

8) TR has complimentary properties to the miner (for the adaptive consensus)
- i.e. not all the transactions shall be sent to all miners

9) TR is executed within the transaction channels (TREVAL), which (channels) can be implemented both in SW (gVM) and in specialized chips (HW)

2.6 FUPE quarks

Если мы хотим по модели смарт контрактов запускать большие промышленные стеки уже существующего в индустрии embedded софта, то нужно рассматривать не только HAL (hardware abstraction level) уровень софта, но и все особенности реализации железной архитектуры ядра (которое запускает контракт).
Т.к. стек embedded софта имеет жесткую связь с реальным железным ядром процессора.

For example, Sirin Labs [] works in this direction.

Quarks

Quarks are instructions of the lowest level of the HW implementation.
I.e. those instructions, on which the contract software works physically.

The characteristic feature of GONT is the displaying of all the quarks of all known processor architectures ***to the intermediate VLIW level***.
For example, the architecture K5 h – the first analog of x86 architecture from AMD – was similarly implemented.

Displaying on the VLIW is considered correct for several reasons.

- We circumvent patents for global processor architectures (for example, for ARM) – as AMD did once.

- The need for wide VLIW commands because of:

1) ONTO Space completeness (512 bits will give unlimited opportunities)
2) Support of the Solidity data width (256 bit)

That's why one can't cope with without the wide VLIW commands.

Herewith, the native code compilation systems should not "notice" the displeying of all the quarks to the intermediate VLIW.

Conclusion (part 2.6):

Displaying the embedded soft at the GONT EVM, we substantially expand the potential market.

3.
What is GONT GAS (AlGas)

The aim of GONT is to integrate into the crypto-economics those business-meanings which earlier were beyond any economics.
(Example: the meanings of the cinema ontologies as a generator of additional cost in mass cinema services).

Services can consist of a variety of micro-services. And micro-services generate work by means of AlGas.
This also leads to the renewal of the Blockchain micro-states, which affect the global Blockchain state.
The service's transfer to a new micro-state is accompanied by the AlGas gas flow.

The set of all the gas "molecules" of one service is the average "token" or currency of this service, which is also bound to the global GONT currency (similar to that how the GDP of a city or region is related to a country's GDP).

Among other things, such a model allows carrying out "ICO on GONT".

3.1 Model of introducing the Tokens 3.0 (GONT-gas)

Let's consider the transaction factors in GONT:

- TR reliability
- TR work (for example, the implementation of a search in Google, with a lot of work, may be TR in GONT) – **see Example 6.2**
- Reward to all the service providers conducting the transaction

- Transaction evolution within transaction channels and intermediate COMMIT states

Difference from Ethereum:
GONT introduces the concept of the complex transaction work.

Properties-factors of the transactions (comparison)

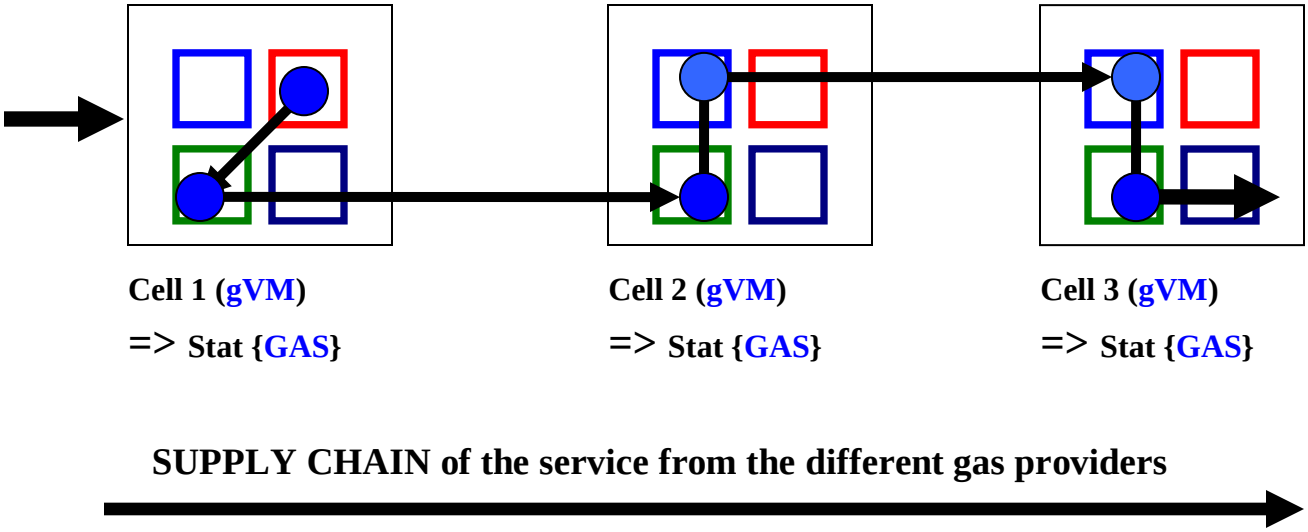
	TRUST (Доверие к транзакциям)	**WORK (Физическая работа транзакций)**	**Reimbuse (возмещение)**
ETHEREUM	High	Low load (простые транзакции)	Модель газа EVM
GONT	High	**Any (high\low) load (транзакции любой сложности)**	Адаптивная модель газа - AlGas (включает в себя модель EVM только как часть)

Herewith, WORK within the GONT conception is **TOW - TRUST of WORK** (proof of the fact that GONT VM of this node really conducts the work and it's possible to request Reimbuse).

3.2 Philosophy of the gas work

Each service in the world has its Supply Chain or Value Chain.

GONT allows building the service's value chain from the alternative gas, which belongs to the different micro-services.



Gas is the equivalent of the work executed by the service. Herewith, the work is "taken" from several providers based on the basic principles of their gas competition.

Supply Chain can consist of completed units of the conventionally "complex work".

The "complexity" of the work can be measured by the debugged code of a product quality or by something else. The main thing is that they are ready-made, debugged modules, which are ready for sale.
For example, the qualitative AR special effect can be a complex gas.
Herewith, "gas" instructions of the EVM and Solidity can be considered a conventionally "simple gas". In the sense of the elementary nature of their work.

The GONT approach greatly expands the philosophy of "gas financial transactions", under which Solidity is grounded.

Complex gas for the complex service work requires a revision of the philosophy of computing capacities.

One of the new approaches on the market for distributed calculations is Fog Computing [SONM].

Herewith GONT interprets the hybrid computing capacities by means of the EVM mechanism, adapting it to a new reality.

It can be said that similarly with Fog Computing, GONT represents a new mechanism for selling distributed computing capacities. But it also conceptually expands the Fog Computing itself.

Expanding the EVM Mechanism.

The expansion of the EVM mechanism has several purposes:

- Introduction of the private ownership mechanism on the Blockchain
- Expansion of the gas space and the gas working space

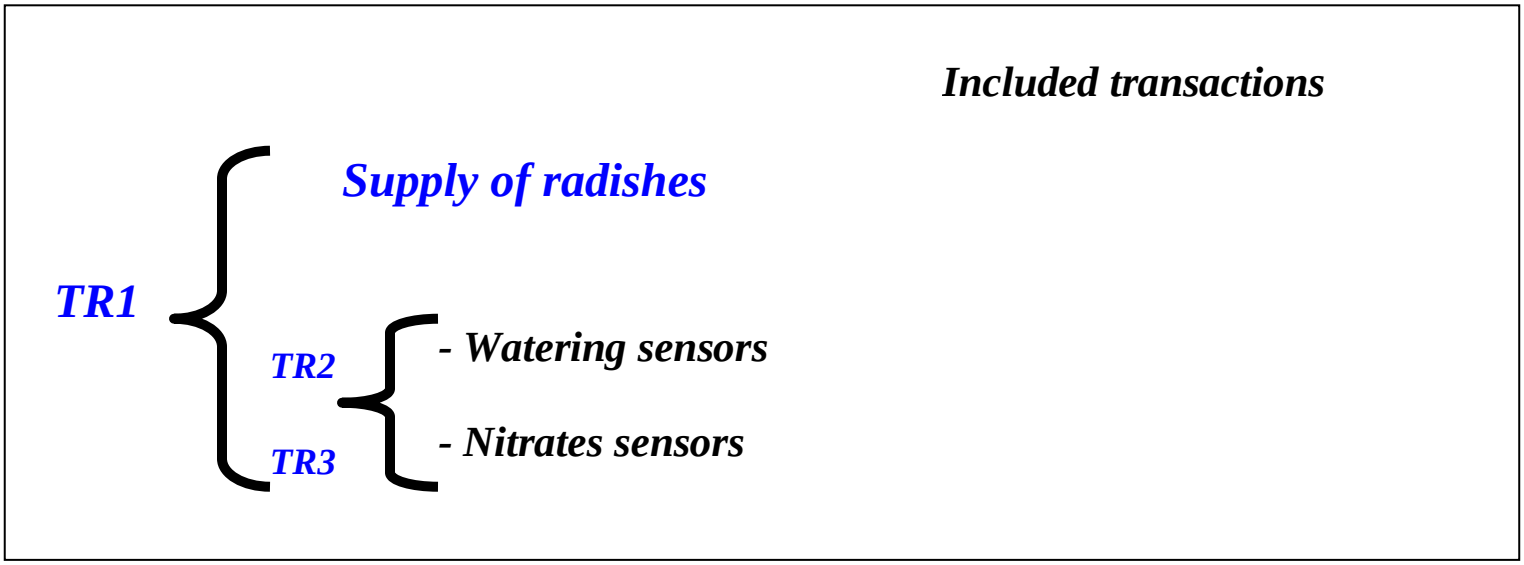
3.3 IoT inclusion for AlGas

Within the framework of the global transactions of the services, there may be intermediate IoT TR from all the possible sensors [Raspberry, etc.].

IoT is a very promising direction for expanding the "transaction market". I.e. it means transition to the state where transactions generally cover the living space of the people.

The GONT philosophy is in expanding "transactionality" through the introduction of new meanings (via GONT Tree) and new calculation methods (via FUPE).

IoT also perfectly fits the concept of AlGas.



Included transactions are described within the GONT Tree

Such included TR are possible within the framework of the VM-decoupling conception offered by GONT.

GONT VM becomes an abstraction combining a physically distributed variety of gVM machines.

The complete Commit for transaction TR1 is collected from the variety of intermediate Commits (TR2, TR3). Wherein all Commits can also be distributed through time.

3.4 Philosophy of the AlGas-based services

The main economic unit of GONT is **micro-services**, but not the services themselves. It's worth saying that exact the micro-services make money.

The meanings correspond to the Micro-services exactly (**GONT Tree Nodes**).

The micro-service can earn money through a plenty of different services, which add the micro-service into their Supply Chain.

The only condition for the service for becoming the GONT participant is the ability to pay for KANT gas – for generation of the gVM (conventional 100 rubles per ontological unit). Since generation of the gVM (fixing ownership rights on the blockchain) uses the resources of the compilation system.

SELF-assembling services

Services on GONT-self-assemblage (GONT builds itself on its own)

GONT gives services not only the "personalized gas", but also data and code containers for gas usage.

Herewith micro-services on generation of the containers are also implemented through the gas (**KANT-AlGas**).

For the moment the KANT gas is designed to maintain the functioning of the system itself and to reward investors/founders.

KANT implements only the basic functionality, not being integrated into the additional cost of the services themselves.

Gas-implication during the container's creation

The container structure is decentralized!
- Advantage over FileCoin, STORJ, etc.

Структура контейнера и сам контейнер могут иметь разных владельцев!
Все это гармонично укладывается в философию GONT.

Dualism

"FUPE-possibility" of the service

Before offering a service – explore FUPE!

The work of the marketing specialist: offering new FUPE types

Expanding the services' opportunities

Example of the FUPE cell within Solidity: []

Example of the Super-FUPE cell within GONT: []

- the possibility of indexing.

Services of the basic KANT

1) KANT1
Search for similar meanings within the external GONT-planes
"Connected similarity"

2) KANT2
Injecting new knowledge into the GONT-Tree (Global oracle of meanings)

3) KANT3
Automatic assemblage of gVM ("equivalent" of the private ownership at the Blockchain)

N) **KANTN**
- lots more.

4.
Advantages for investors

4.1 Services

The real services of the real world will stand to the maximum benefit from GONT.
And this is not only the opportunity of the quick service assemblage in the design mode but also an automatic gateway within the Blockchain.
The most important thing is the opportunity to earn money for those who didn't even think of becoming a part (one of the providers of their own knowledge) of a global service.
After the introduction of the "service gas" each creator of the service becomes "bit an owner" and continuously receives a profit from the service.
Example: chess24 and film company.

GONT and Blockchain will provide absolutely accurate billing on the use of distributed intellectual property of the service and will automatically distribute the reward. Thereby we'll cover a huge variety of service websites.

4.2 What makes GONT unique for connecting real services

- adaptive consensus for services. Take as much consensus as you really need.
The routing layer gives an opportunity for dynamic distribution of the "consensus" to the groups of miners.

- GONT simplifies inventing a "token economics" for the services conducting ICO.
AlGas makes the economics of the token work very vividly for almost any business idea.

4.3 Gas as a systematic basis for the stable currency

The main gas property is the accurate measurability of the involved resources of the decentralized network, and therefore the accurate billing.
This leads to the transparent (for all the participants) basis of the physical background of the service's prime cost.
It's worth noting that tokens don't have such a property.

4.4 Stable currency

The currency based on the balance of issue s and the work of the service gas will be guaranteed protected from collapses and cost jumps. And that's what is attractive for large funds.

Balance as the essence is based on **the continuous self-modification of the GONT-Blockchain**. Adding a new service leads to the generation of a new "private ownership" within the Blockchain (at the level of the Blockchain's initial code), which ("private

ownership" of the service) will affect the direction of the Blockchain's development (growth) through the generation of a new type of work ("additional cost") and topology of the network with an internal transaction routing map, and suchlike.
I.e. the certain feedback mechanism appears in the system, as in a normal self-regulated stationary economics.

All these things will give an opportunity for applying the adapted macroeconomic formulas within the Blockchain network and for forming the cost of the crypto-currency for the open market.

In fact, the currency cost occurs during the "burning" of the service gas through simple and complex service transactions from the end consumers of the service mantle.

GONT currency doesn't have a center influencing its cost. The cost consists of the gas cost of each service, normalized to the GONT-economics' balance, and its actual flow for the period. In simply words: **the need for purchasing new gas causes the need for issue of money and vice versa**.

5. GONT economics

5.1 Advanced gas comprehension

Aim of the GONT-economics:

Creation of the technology for the possibility of integrating a "stable" currency for mass services.
Basing on a stable currency, laws of macroeconomics (crypto-economics) money will work, which will invoke even more significant investments into the crypto-currency.

We introduce the alternative service gas (**GONT AlGas**) as a need to measure the services' work and accurately calculate the network's GDP (cumulative gas burning by all services). And also as a private ownership mechanism in the Blockchain network.

By means of GONT one can move from the empirical rewards of the miners within the current Ethereum network to the balance cost of the currency reflecting the economic reality of the network, based on the services' physical work.

5.2 Issue model of the GONT currency

What distinguishes Bitcoin from Ethereum?

Ethereum has no issue limit!
This means the possibility of two types of issue : positive and negative (burning of the currency).

Models for GONT

GONT also has no issue limit, but issue should be regulated.

Let's introduce the concepts

Primary GONT economics

- Variables of the primary GONT activity, for example, generation of the gVM cores for the services' maintenance. But the gVM generation itself doesn't mean that the service has attracted some users. Service users are displayed in the variables of the real GONT economics.

Primary variables represent to the investors a general economics' warm-up (the number of the warmed-up and potential customers). Primary economics is like the requests for utilization of the real economics.

Real GONT economics

- Variables according to the services' launches and service gas flow. This directly affects GDP and issue.

Smart-contracts

All the GONT economics objects are regulated by smart-contracts.
Example of the currency regulation by means of smart-contracts and contracts' audit:

Terms
GCB - GONT Central bank (contract of the Central Bank)
SH - Stake holder – miners who hold some currency volume.
Theoretically, as usual, anyone can become a miner. There'll be only requirements to the quality of mining-services. For example, at the data center level and not lower.

5.3 Issue generic ideology

Cycle of the issue (positive or negative) is **one week**. In fact, this is a cycle of basic statistics on gas flow throughout the GONT system.

The issue is calculated in reference to the forecast of growth of demand for service gas for a week ahead. The forecast is built automatically from the data of the previous weeks (under one of the models). It's expected that there's a continuous increase in users and services to the system.

Automatic feedback and issue regulation

The main question is – how to burn the GONT currency if the demand for gas has decreased?

It's obvious, that miners won't be in any way interested in the voluntary burning of the currency.
Our main aim: support of the regulation **Currency volume = service's volume**
Herewith GONT currency can be always sold at the stock exchange at a more or less stable price in FIAT.
If the currency volume is less – deficit demand for gas occurs. There'll be no possibility to pay for transactions.

If the currency volume is more – deficit demand for dollars (FIAT) occurs.
Users will have problems with the crypto-currency withdrawal. This can cause the avalanche effects of currency dumping and panic.

The ideal case of balancing the entire system is working without stock exchange, but through the ***guaranteed GONT-FIAT exchanger***. It guarantees at a fixed price (reference to the electricity price) an exchange in any direction.
The stock exchange can correct some "screw-ups" of the system balance.

5.4 Model of the GONT Central Bank and negative issue .

Let's go back to the main question – how can one guarantee "burning" of the currency along with the reduction of demand for the service gas?

Terms

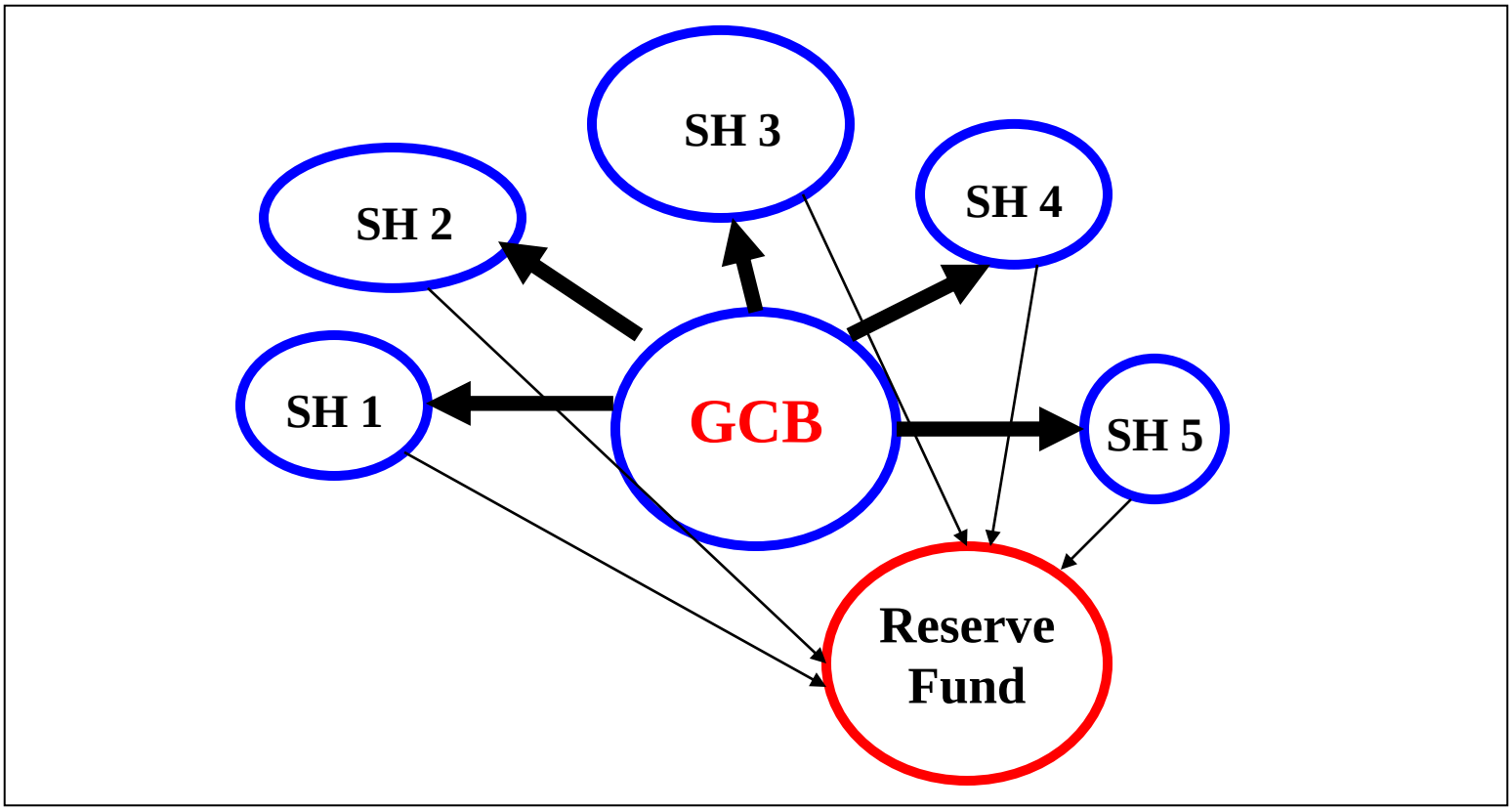
GCB - GONT Central bank (bank smart-contract)
SH - Stake holder – miners who hold some currency volume.

GCB (bank contract) is the only issue source.

GCB distributes the SH currency for the executed work (on hosting and mining).
But the currency is distributed not just for fun, but on the conventional credit at a conventional "re-financing rate".
The re-financing rate is needed for currency withdrawal from miners for potential burning.
But this mechanism is of the rewarding character.
The withdrawal is performed automatically (at the "re-financing rate") at the moment of conducting payments on the rewards to the miners into the special ***reserve fund***.

If the gas flow within the entire GONT continues to grow up, the currency is not needed to be burned and it's returned to the miners at the next cycle (full reward of the miner along with the economics growth).

If gas flow reduces, the ***funds of the Reserve fund are burned by the system***.



PoS economics of GONT

Pre-GAS economics (primary) – generation of the GONT cores.
Post-GAS economics (real) – generation of the cores' work within the services.

5.5 GDP of GONT economics

Main parameters of GONT GDP:

- Total gas flow within the entire GONT economics. And also gas flow calibration in Joules and transformation it into the FIAT cost.

- Amount of the services' launches.

Herewith each GONT participant can independently verify the gas flow (basing on the Blockchain records).

If the WoW (week over week) gas flow doesn't change, the issue will be zero-point.

Here one can think about ***the minimum required issue*** for compensating to the miners the costs for maintaining the system.

6. Technology

6.1 On innovation

There are several projects extending EVM space (**RSK**, **Sputnik VM**, **nebulas**.io, etc). EVM extension can be insignificant and rather nominal (RSK simply introduces new macros from combinations of previous commands) – for operational comfort, but it can be radical as well - for implementing new classes of tasks that are yet to enter the market.

GONT VM is a conceptually new approach to building EVM Blockchain.

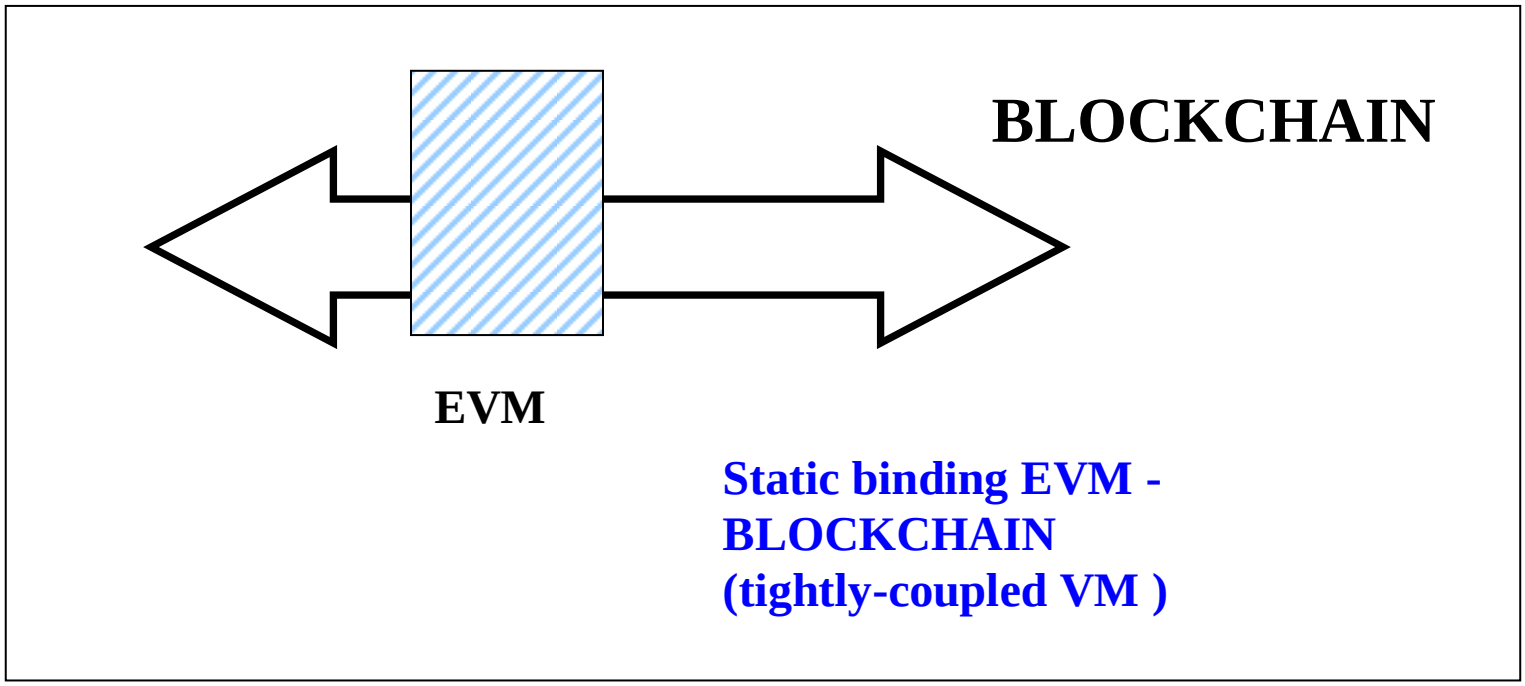
- Sharding for EVM (implementation in GONT VM)
- Multi-cluster EVM (implementation in GONT VM)
- Increased safety of EVM (introduce TLB and **TransactionsTrustZone** technology)
- OS support capability (in one of GONT VM clusters)

6.1.1 On implementation of Plasma.io

GONT aims to offer the implementation of Plasma.io technology and philosophy, among other things. At the same time, GONT extends Plasma.io philosophy

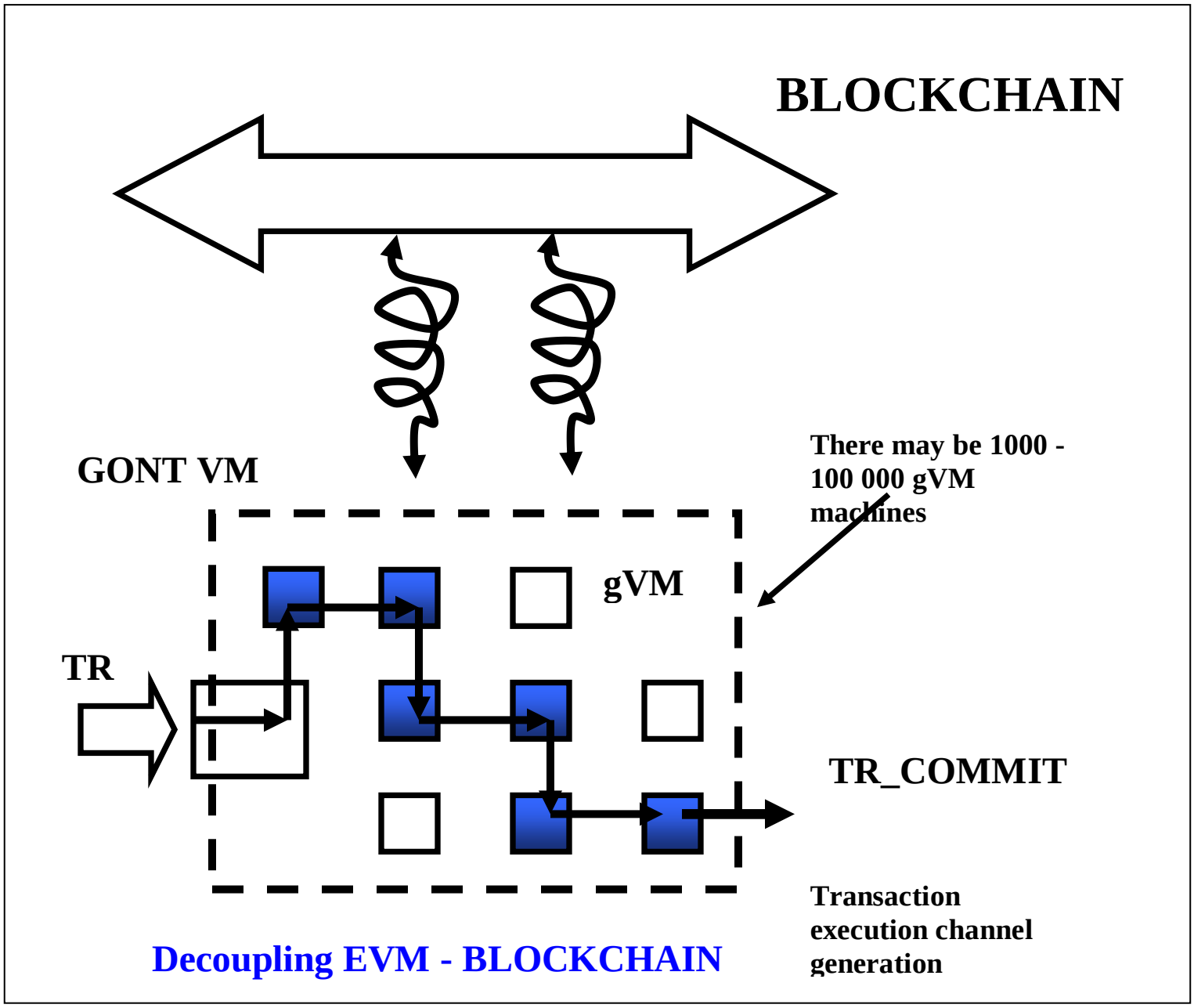
All blockchain objects are subjected to sharding in GONT.

BEFORE (Normal Ethereum):



NOW (GONT) :

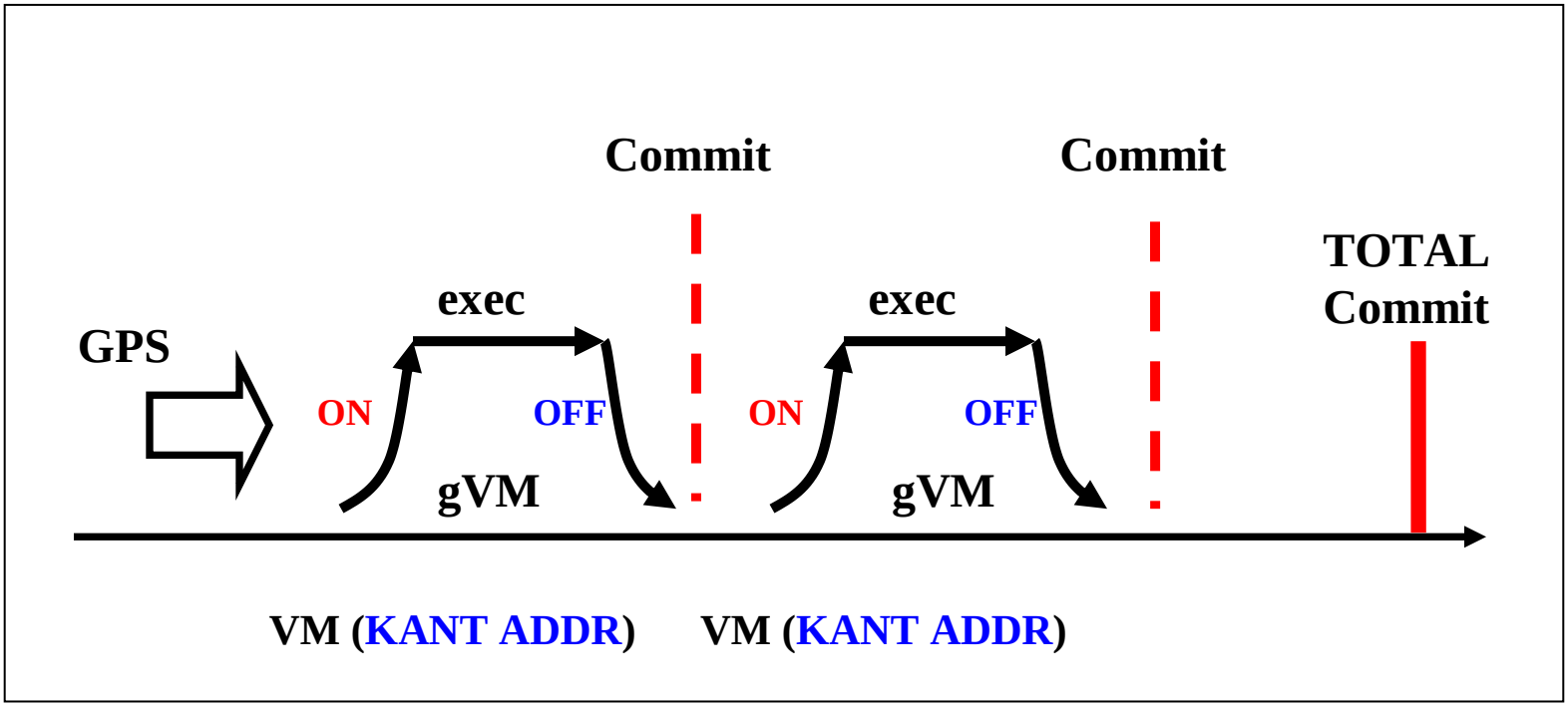
EVM-FAB: A factory of virtual machines with internal routing of states.



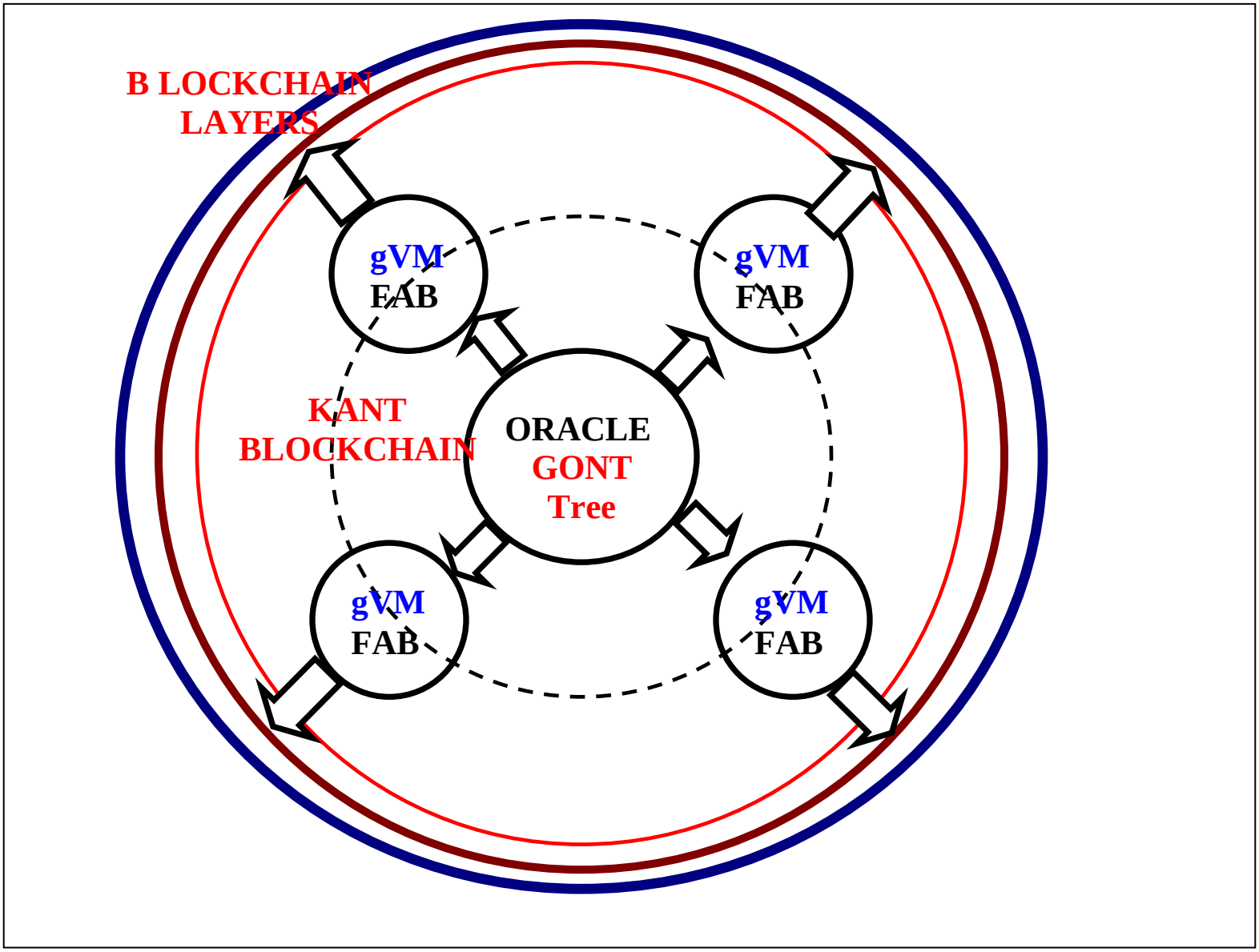
EVM structure is changing dramatically (GONT VM implementation), while remaining fully EVM-compatible.

Pattern of gVM chain execution, through which TR_STATE passes

The gVM machine pool can be very large, the appropriate machine is turned on only for transaction development and then it is turned off again.



KANT ADDR addresses gVM machines for routing and selects the required VM from the gVM pool.



Blockchain layers from Plasma.io

KANT BLOCKCHAIN describes the integrity of the GONT assembly (for both gVM and ORACLE).
- There is no centralization around ORACLE!

6.2 Extended GONT-interpretations of blockchain entities.

6.2.1 Extended interpretation of STATE

GONT interprets STATE VM and STATE transactions separately

State is a prime attribute of Ethereum.

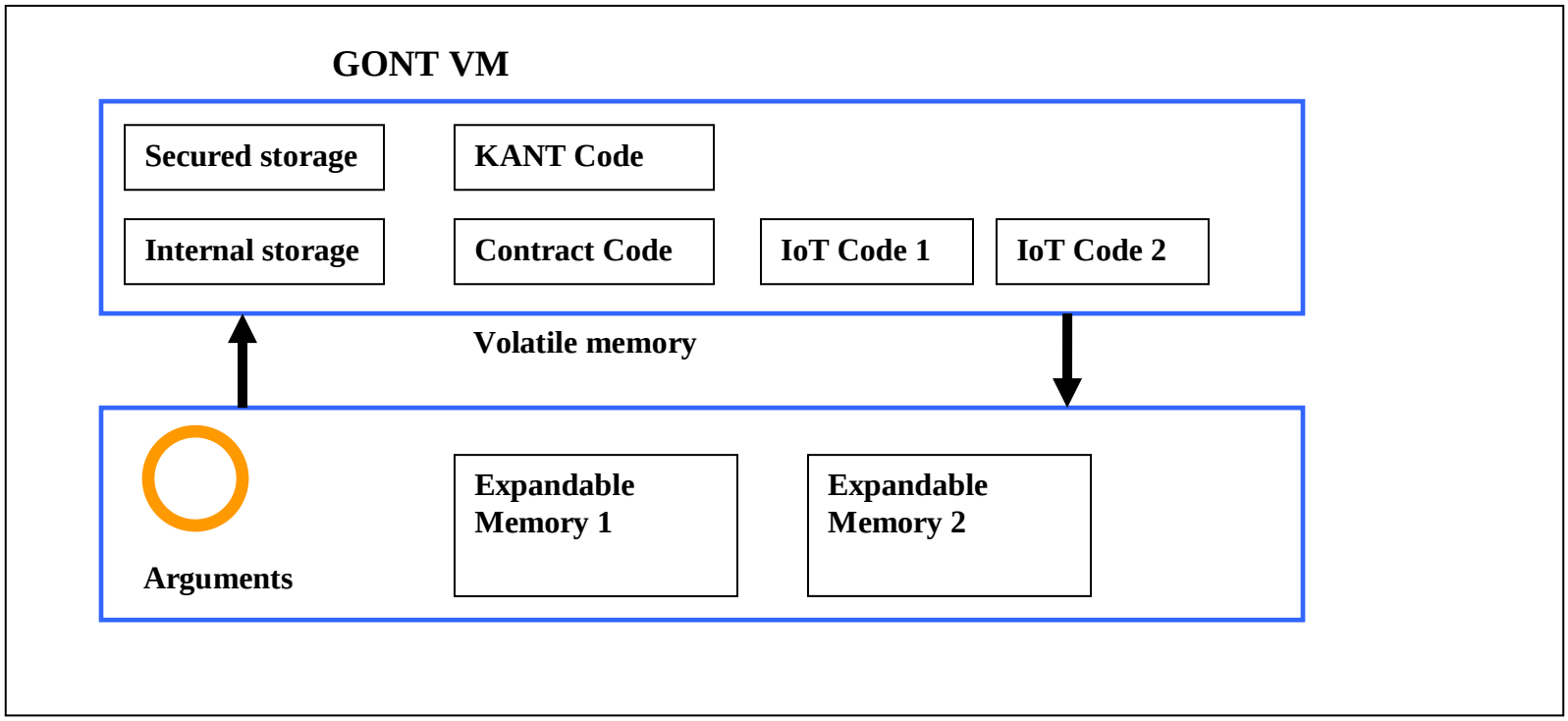
State was introduced by Buterin for storing the results of EVM work (or storing EVM snapshot while developing the contract code)

An obvious innovation for implementation is to expand the understanding of STATE significantly, at the same time preserving its “economic compactness”.

Visualization

Example for GONT-IoT services

- State is extended by the code repository for Embedded devices
There may be a code store for each cluster inside gVM.

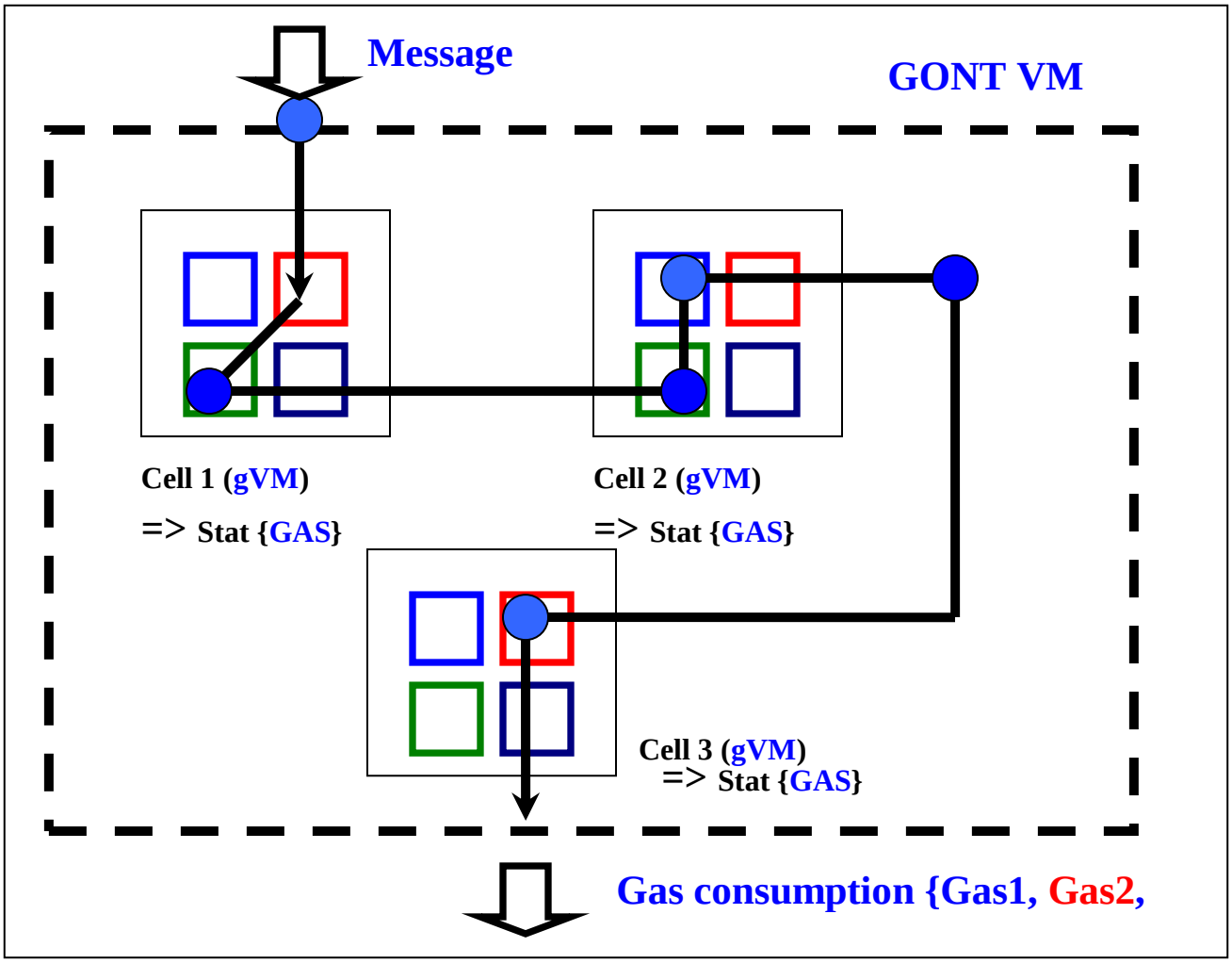


6.2.2 Extended interpretation of EVM

gVM and GONT VM

gVM comes with a primary knowledge of the external service (pre-compiled contracts) - unlike EVM Ethereum. EVM knows nothing about the service. Pre-compiled GONT contracts are little or no different from EVM Solidity contracts. In both cases, there is almost no contract change space after deploy.

Concepts:
GONT VM is a global virtual blockchain machine.
gVM is a single micro-machine inside GONT VM.
Cell is the bound of the ownership of one user inside GONT VM. Cell is displayed on ONTO slice in GONT Tree.
GPS (**GONT Processors States**) is a new routing protocol inside the GONT VM boundary. Routing of external messages among gVM cores



Implementation of VM sharding.

GONT VM is divided into a great number of gVM. The number of gVM cores is not limited yet.

After having developed each core in each cell, we get the statistics of the service gas consumption at the output.

An approximate number of loads per gVM core is about 1000 - 10 000 gas meanings (**1000 - 10 000 GONT Tree nodes**).

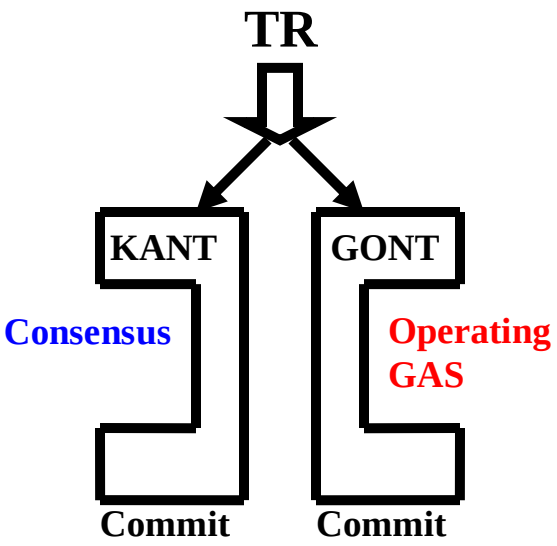
6.2.3 Extended interpretation of consensus

Gas and adaptive consensus

Computing in the FUPE space may be very expensive for complete consensus (for example, the gas function may be a complex pattern recognition function).

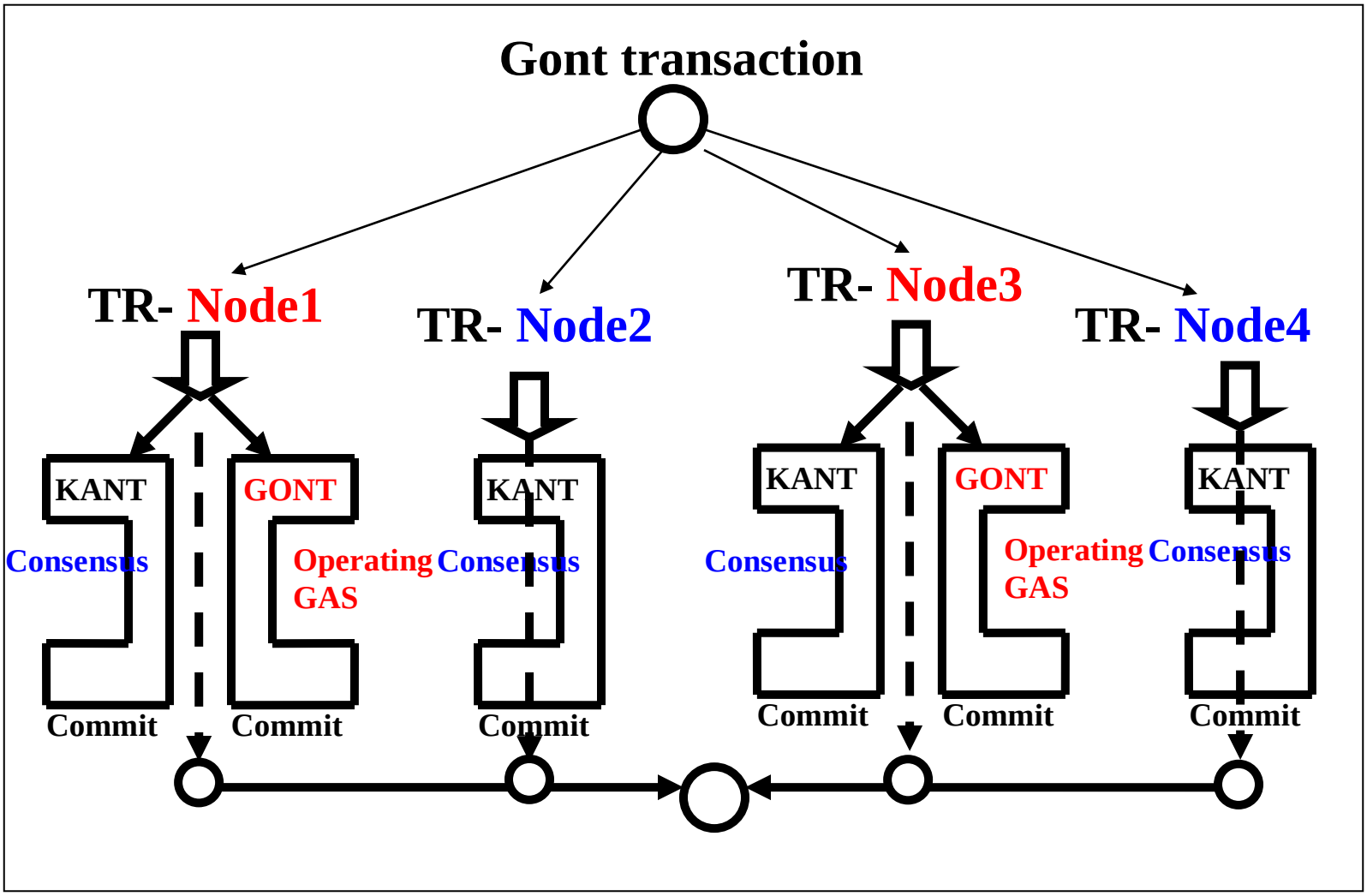
A reasonable solution is **to let the service choose the level of the required consensus** (for example, it can differ greatly in case of financial transactions and recognition functions).

In this case, the transaction is split into two execution paths. KANT Path, which is executed to the maximum (maximum consensus) in any case. And GONT Path is executed only in FUPE space. Thus, all nodes find out about the execution on some random (or selected ones by the service rule) nodes of an ultimate GONT-FUPE transaction.



TR is split into ultimate consensus (KANT path) and ultimate gas operation (GONT path).

Commit is the status of a successful execution of a transaction or a part of transaction.



Adaptive GONT consensus

Adaptive consensus suggests operating gas to be run not on all nodes, but only on some of them, namely the ones selected according to a specific rule (for example, by random sampling).
It means that there is no economic sense to run complex operation on all nodes at the same time to prove execution of TR.

7.

GONT contract execution pattern (v 1.0)

General approach to the development of the contract code execution pattern

Implementation options:

- Without **LLVM** (a simple compiler)
- With **LLVM** (a complex optimizing compiler – similar to the one for Solidity) [2]

For now, let's consider the option without LLVM and without entering GOL functional language
Without LLVM – a simple compiler (actually, it's just a generator of ELF file for execution in the gVM core).

GONT implements and supports both full backward compatibility with Ethereum (Solidity contracts) and its own contract pattern.

7.1 Contract

What is GONT-contract (v1.0)?

- Total number (any number) of transactions (TR) from descriptions on GONT Tree.
The contract is always in the state of one of TRs, then it passes to the next TR under the influence of external parameters.
This pattern is similar to "The smart contract language in ZILLIQA **[1]**.

7.2 God-given contract evolution

GONT- smart contract can be considered as an evolution among the essential states of the program-contract (the evolution of meanings).
Evolution follows from the meaning of economic activity, which is described in the contract, in a usual, native way.

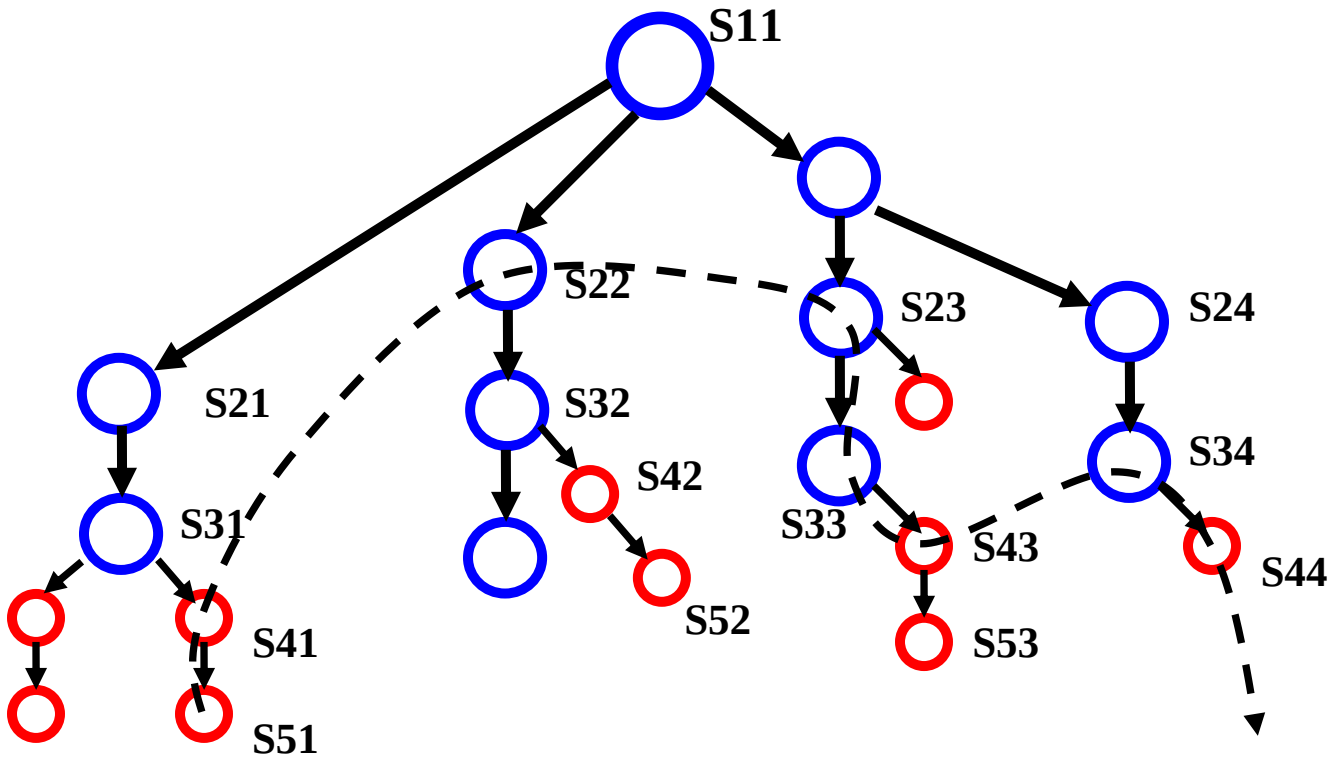
This definition of the contract follows **from the way the AlGas gas is entered into GONT**.
When you enter AlGas, any transaction "turns" into a processor (gVM core) instruction and is one of many instructions of the global "gas".
At the same time, the instruction describes some meaning, specified by the user.

Transactions are entered through the oracle tree "**GONT Tree**".

In visual mode, the user can select “program meanings” on GONT Tree and specify the contract evolution, actually - FSM (Finite State Machine) of the program - in precise terms.

7.3 Signature (Sig)

As the contract “passes” through the states, a signature is formed, which is the object of consensus on Blockchain. Consensus can be entered in different ways. The signature provides an object for the comparison formula on different miners.



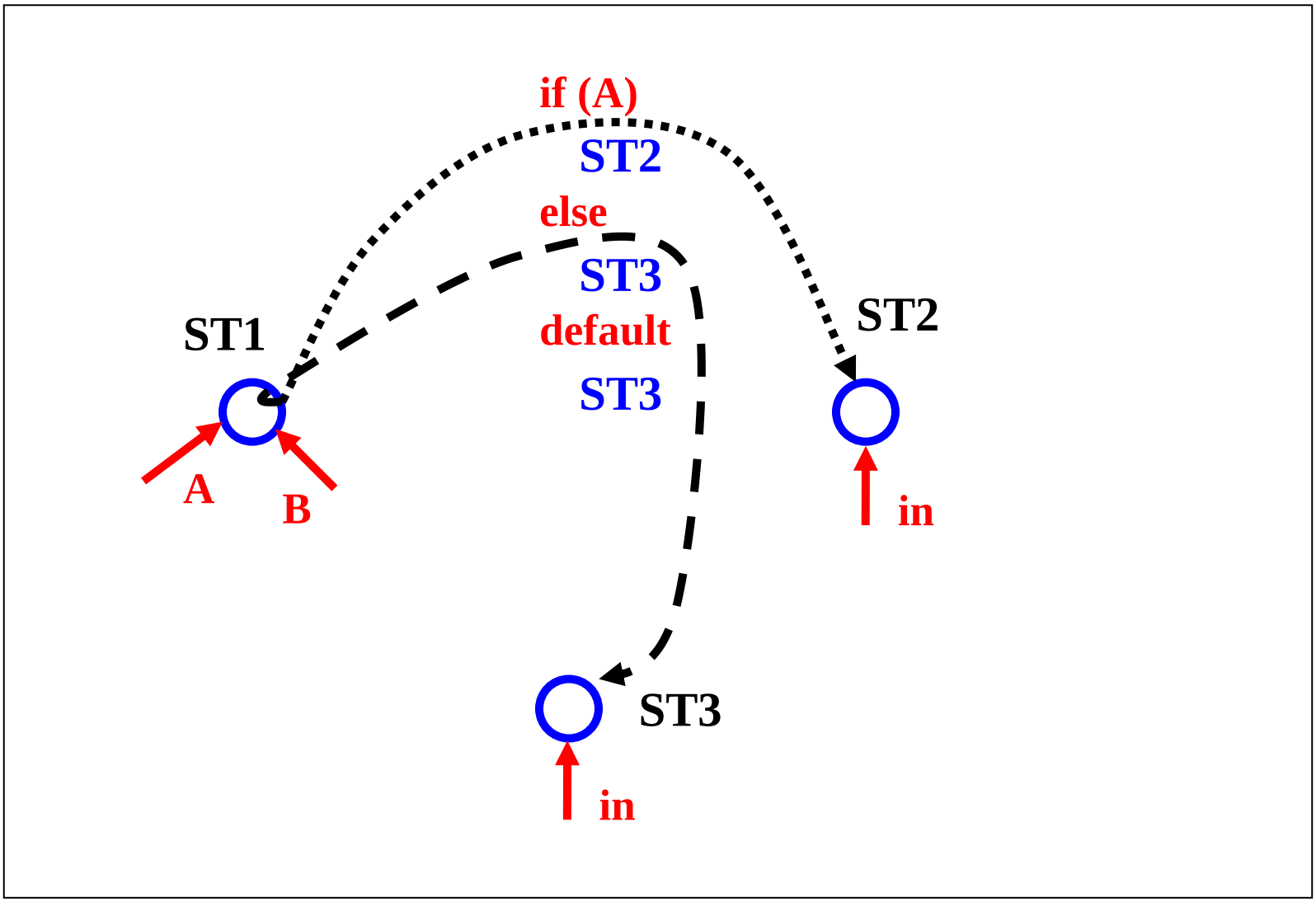
Program code (signature):

INIT	S51	S41	S22	S23	S33	S43	S34	S44	END

GONT Tree is the ontology of transaction meanings. Contract program is transitions among states of meanings under the influence of external conditions.

The advantage of the approach is the possibility of "visual" (simple) programming without profound examinations in the language of contract writing.

7.4 Contract execution pattern. Scheme



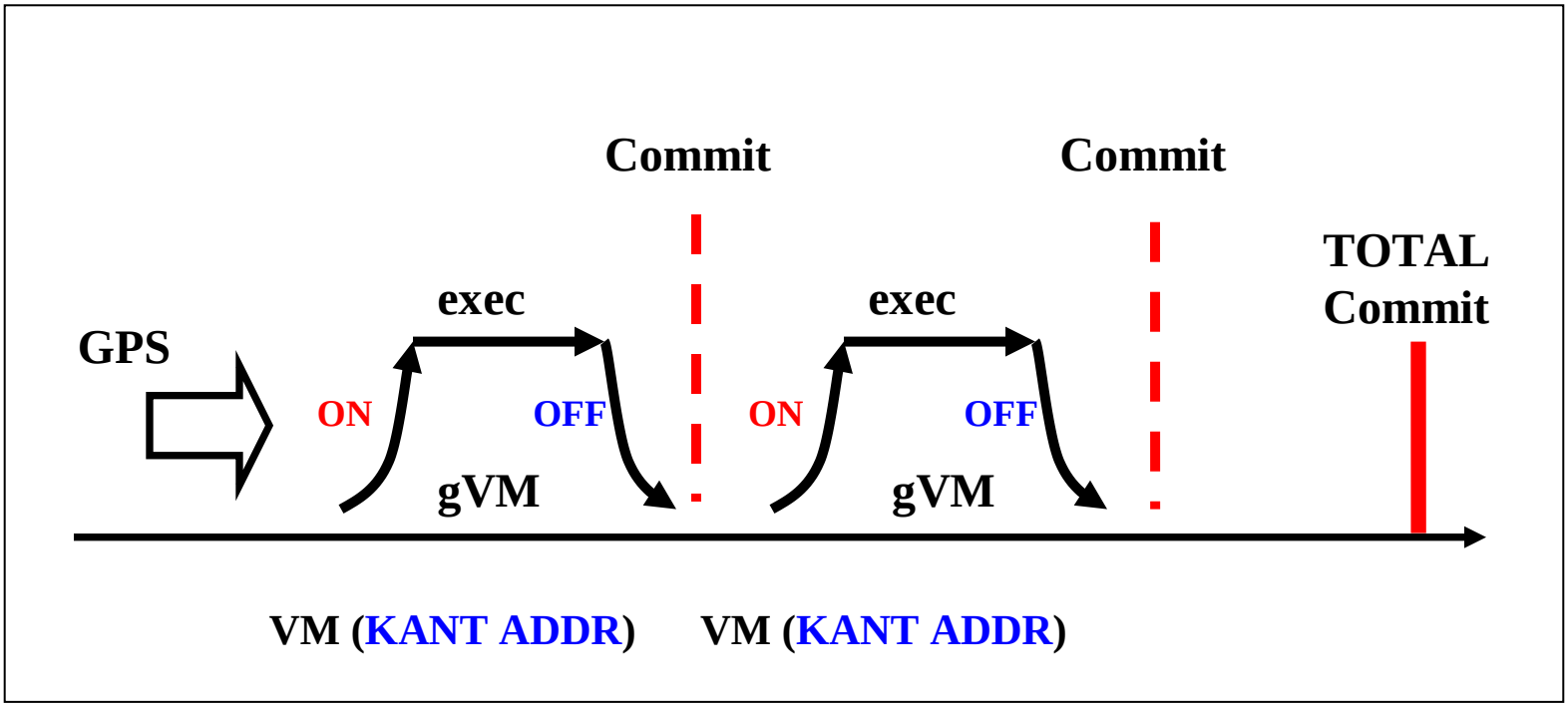
Pattern of a single contract instruction execution

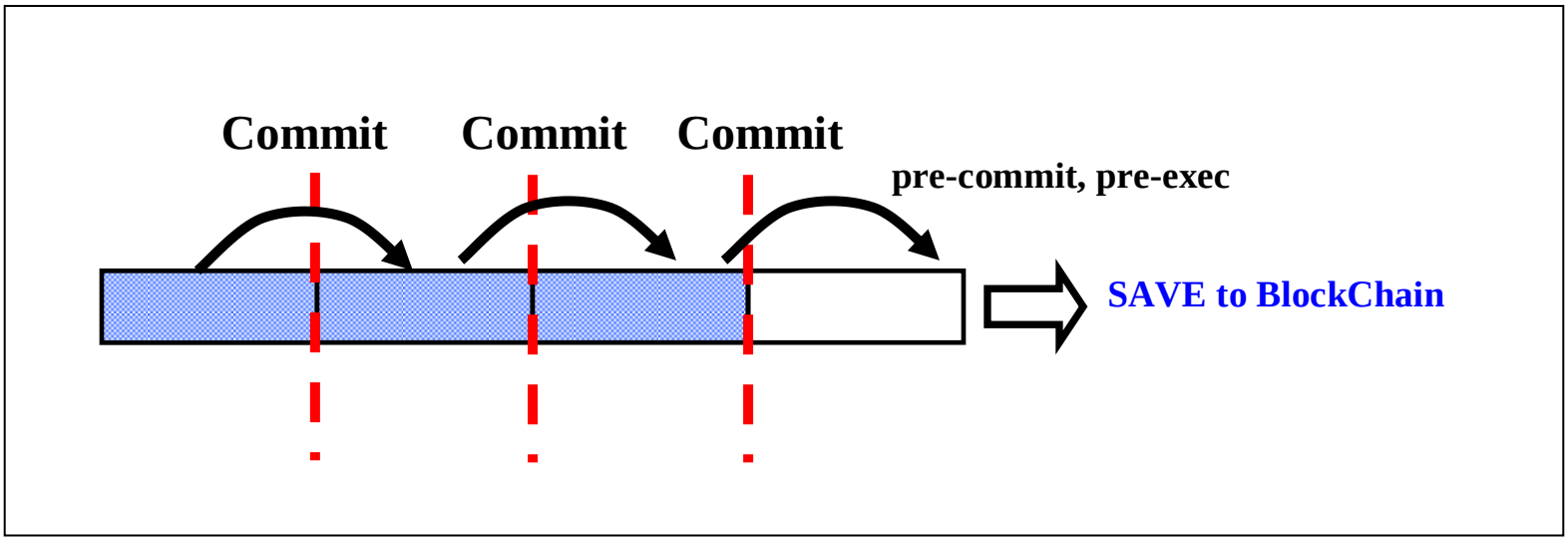
Pre-Exec pattern

After the current instruction is executed and the **Commit** state is reached, the signature must be extended (by self-modification) for setting the next instruction, specifying where the state will pass.

Physically, this instruction is written up to the signature, but with the "**Pre-Commit** And **Pre-Exec**" flags. It means that the instruction has not been executed and confirmed yet. After this, the signature state can be saved in the blockchain block.

After resuming execution of the signature (at any time), this instruction will be executed and extended with a new instruction in a similar way.





7.5 Sigs and Web services

Simplicity of implementation of Web services. **Hybrid Off-chain - BChain approach.**

GONT offers an approach for implementing Web services, if their interaction with blockchain is necessary.

In case of normal operation, the site implements a huge number of states (which are developed by MVC controller), and not all of them should be "reflected" in the blockchain.

When the Web service starts, the states, important for the blockchain, should be considered separately.

At the same time, the web service is considered as a machine of states (**FSM + state set + evolution pattern**), which "flow" among each other under the influence of user behavior. However, not all site states can be helpful for recording them in the global blockchain.

The sig describes the evolution of the state of the micro-service from the launch of its local Genesis block (the beginning of the micro-service operation), constantly self-modifying itself (self-modifying of the sig by the gVM core) by adding a new state to the end of the sig (according to the evolution pattern).

The concept of a proxy sig of a service.

As Actions and Bactions are executed, the service completes the service sig.
The sig should be synchronized on the service side (OFF-Chain side) with the sig, saved in the blockchain block.
After verification and synchronization (according to one of the algorithms), the sig can be considered as proxy and ready for the next command.

7.6 Bactions u Actions

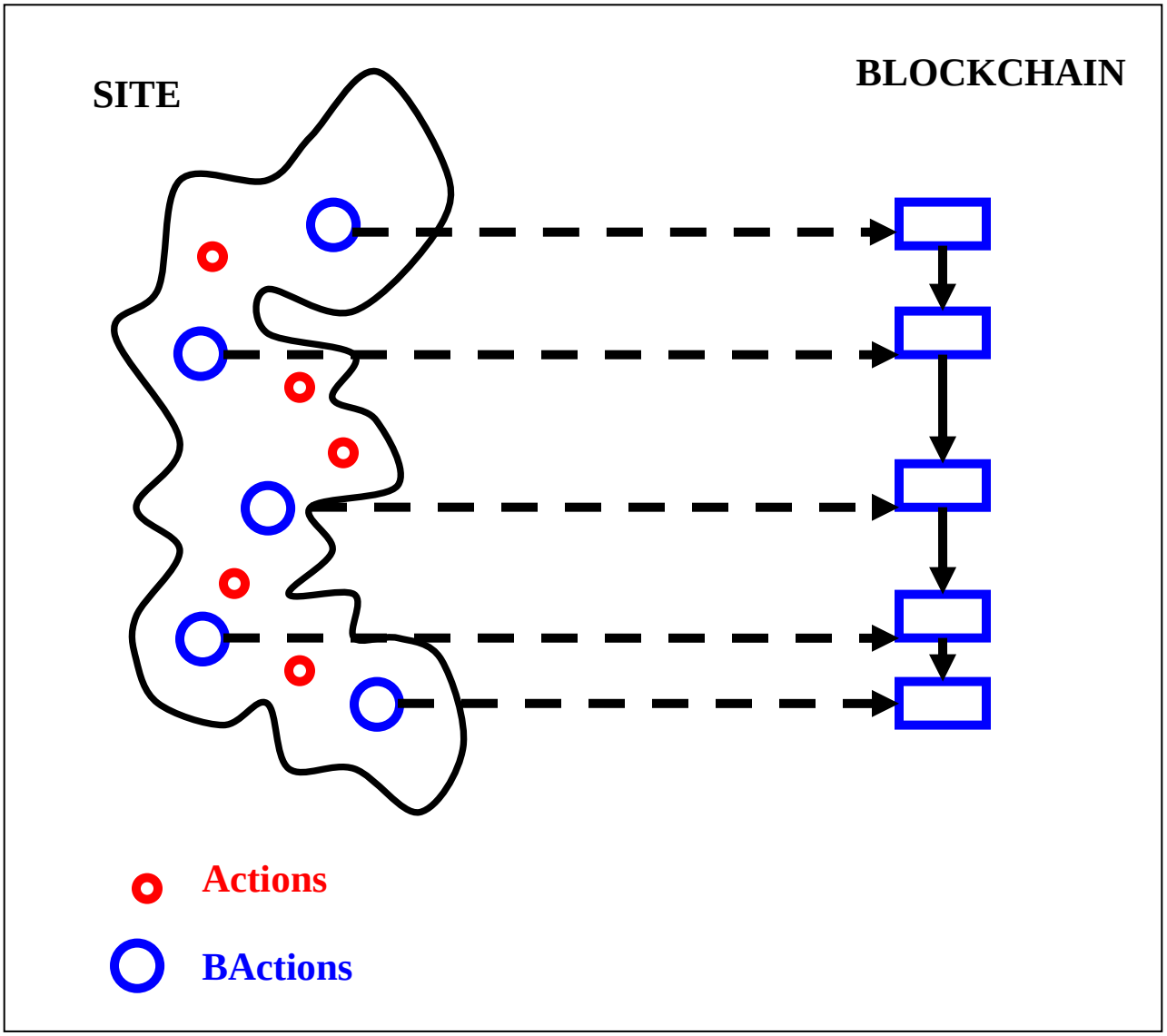
===
The general methodology for all **GONT Web services** is not only for Elections GONT service.

State Machine – Site state machine

The user may perform a lot of **Actions** on the site.

But some of these actions have significant meaning peaks and should be reflected in the blockchain (important transactions).

At the same time, it is necessary to align the site **state machine** and the corresponding site blockchain service. It means that a proper reflection of the site should be built on the blockchain.



Our approach is related to Buterin's in **Plasma [plasma.io].**
It means that we are building a consensus pattern for embedded blockchains.

The implementation is difficult to compare because the implementation of Plasma has not been published yet.

Implementation through GONT VM

In order to reduce the resource consumption, GONT VM doesn't work continuously.
In fact, GONT VM is turned on, executes new transactions and is turned off.
Besides, the last GONT VM state is saved in the service signature (it is a local GONT BlockChain as well).

==
Let's enter **BActions** = **Blockchain Actions** – Actions of the user for Block chain.

Thus, we get **the state evolution pattern**.

Example of the evolution pattern for Elections

BActions:
- INIT
- Registration
- Vote {Vote options}
- Stamp
- WriteBack

Evolution:

Each BAction has a dedicated unique address in GONT Space.

Conclusion: when developing a new service, **Evolution and Actions** should be elaborated accurately.
Evolution is implemented through a service smart contract in the core of GONT VM.

Affecting documents (part 7)

1. The smart contract language in ZILLIQA
https://drive.google.com/file/d/0B1GfJTgMudzgX1d4RS1TUlJCVEE/view

2. Ethereum LLVM
[https://github.com/ethereum/evmjit]

8.

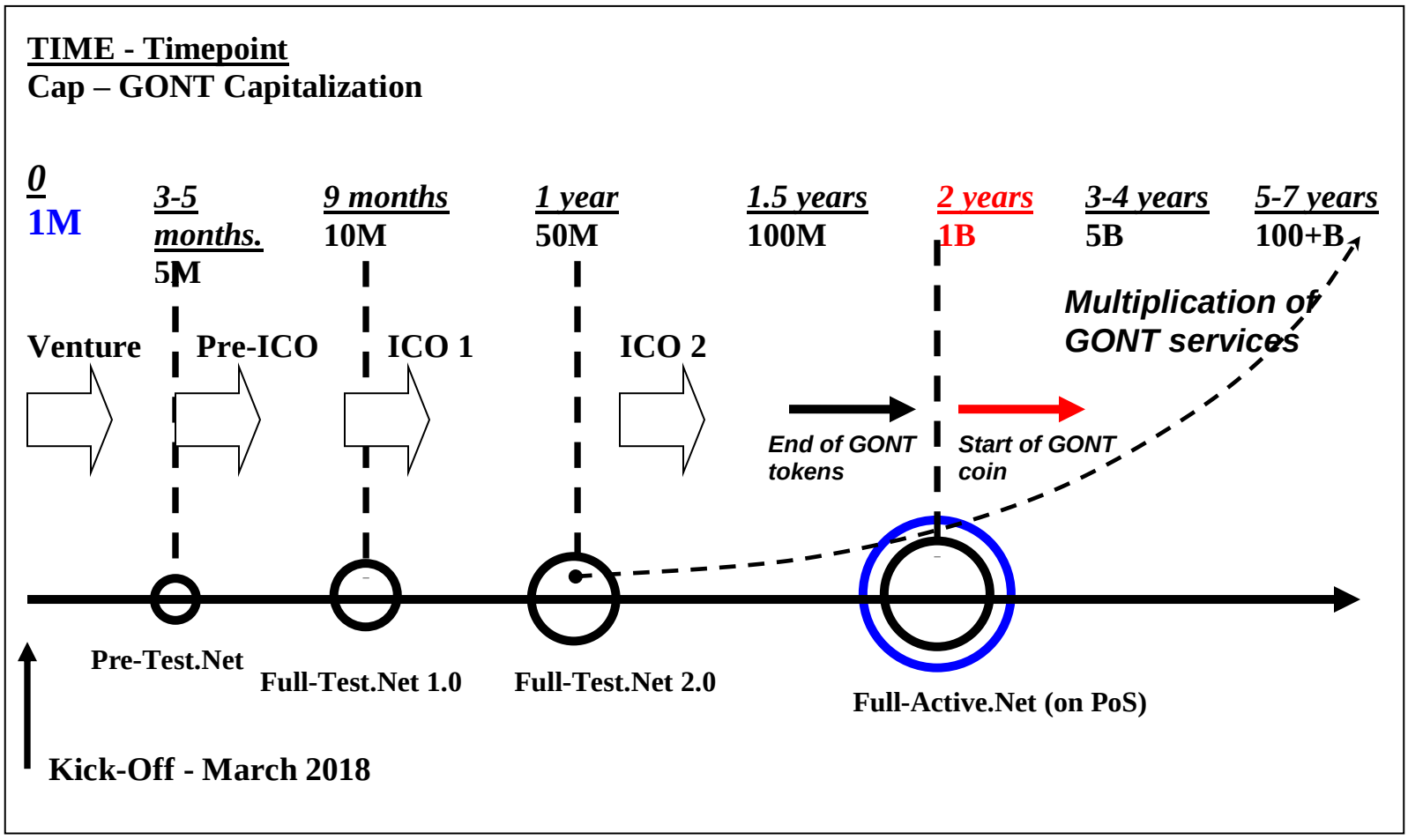
Evaluation of GONT capitalization and its development stages

Introduction

Powerful information and technology reasons (such as launching its own functional GOL-language, compiler release, Test.Net, Active.Net, etc.) are developed for each upswing in GONT capitalization.

GONT is the only project in the world that works with the **real cost** of tokens (service tokens). Technologically, this is implemented through AlGas and is the fundamental basis for the growth of GONT as a whole.
Classifiers and growth drivers will be considered further.

Let's consider timeline of the capitalization growth and fundamentals of the growth.



8.1 Capitalization growth stages

A detailed SWOT and Risk Management will be performed at each stage (in the course of operation) and the growth review will be shared with investors.

1. **Kick-Off - March 2018**

- Capitalization : ~ **$1-2 M**

Assets: There is a prototype of gVM core (as a substitution for EVM in Ethereum) and blueprints of many future services. A prototype of the first GONT service – "**GONT.Elections**" – which is a service for elections and voting, was developed on this gVM core.
Moreover, there is a technology for automatic generation of the gVM core from meta-descriptions of the processor commands on the oracle. We think that this is enough to start a global project in terms of technological groundwork.

Capitalization is based on the assessment of the prospects and on the profound study of the topic with a horizon of several years.

Usage of the technological "lever"

GONT will be launched on the basis of "great predecessors" – Bitcoin and Ethereum – and, as a lever, it will use all the technological achievements of Ethereum while passing through the timeline stages. In order to protect the investment of all players at the crypto-market in solutions and software on Ethereum, GONT will be backwards compatible with Ethereum at any timepoint. At the same time, GONT will offer significant improvements as well.

2. **3-5 months (after the start)**

- Capitalization : ~ **$5 M**

Rapid growth of capitalization is related to a significant speed of prototype developing (**Pre-Test.Net**), based on the existing significant developments in the gVM core.

Besides, the activation of marketing should help in creating a full and clear picture of growth prospects.

Moreover, the pipeleine of prototypes for 100 real GONT services will be developed (are already being developed) before the launch of **Full-Test.Net 1.0**.

The start of Pre-ICO carrying out that will provide us with the first real market evaluation of capitalization.

3. **9-11 months**

- Capitalization : ~ **$10 M**

Assets: Launch of **Full-Test.Net 2.0**

Pre-ICO has already been conducted and efforts have been fostered to the improved prototype Test.Net, which will be demonstrated at the ICO stage. The prototype is demonstrated to everyone as an open asset **Full-Test.Net 2.0**.

After the launch of **Full-Test.Net 2.0**, it is planned to launch 1-2 combat services on GONT per week (on the test network). From this point on, the capitalization process will speed up significantly.

The operation of the services will be actively demonstrated all over the world. Load testing will be used for economic planning of scaling.

The start of Pre-ICO carrying out that will provide us with a new real market evaluation of capitalization. Presumably, at least $ 10M.

4. **1 year (12-15 months)**

- Capitalization : ~ **$50 M**

Assets: Launch of Full-Test.Net 2.0

Approximately at this time, theoretically, GONT will be able to profit off of AlGas (regardless of profiteering).
KANT service gas system will be launched, which will enable to charge the users for business transactions (for example, for automatic assembly of gVM) or for injecting meanings into GONT.Tree (Oracle of Meanings).

From that time on, active filling of GONT services will begin. Part of the services will operate in the test mode, and the other part will run in the combat mode.

5. **1.5 years (18-20 months)**

- Capitalization : ~ **$100 M**

Approximately at this stage, the second ICO will be carried out. It will confirm or disprove the current capitalization and add resources to overcome the capitalization of ~ **$1 B**.

6. **2 years (24-30 months)**

- Capitalization : ~ **$1 B**

Exceeding a billion dollars capitalization lvel.

- Based on a moderate growth scenario. Theoretically, it is possible to capitalize $1B much faster (see EOS). We want to link the capitalization growth pattern more to real services on GONT, and less - to technical methods (pump, etc.)

Assets: Launch of the operating PoS network as an Ethereum fork and with the extended fully functional GONT architecture.

50-100 services are operating in combat mode in the network (GONT mantle).

All businesses get to GONT from the real world, as to the vortex of the whirlpool.
The extended concept of GONT transaction capability enables to cover the maximum number of businesses (from financial payments to IoT, booking business jets and growing radishes in the fields)

50% of businesses come voluntarily.
50% - are attracted by the prospect and benefit for the businesses themselves.

GONT becomes total.

All tokens previously released on ICON GONT cease to exist and are converted automatically into the PoS GONT currency (with a bonus for “consent”)

GONT becomes an independent blockchain with its currency and economy.

6. **3-4 years**

- Capitalization : ~ **$5-10 B**

Assets: the release of the first specialized GONT servers on GONT chips (or FPGA).

GONT tops TON-Telegram in capitalization.

7. **5-7 years**

- Capitalization : ~ **$100+ B**
capitalization is already considered to be the total capitalization of all services on GONT using the GONT coin.

GONT state:

A thousand real services operate in the “business without owner” system and “business without mediators” system. The whole system is finally decentralized. Each participant fights (competes) for his/her PoS stake (for its growth).

GONT is finally decentralized and starts free sailing, independent from the founders (like Bitcoin).

The protection against “mediator attack” works to the fullest. Nobody can become the mediator for services.

None of the states can take GONT under their control and begin to manage it.

The system generates enough AlGas transactions to be noticeable for the region’s energy balance.
The stability edge effects start to occur. They should be solved by a transition to a qualitatively new level, that is, optimization of service calculations at the level of specialized GONT microchips.

The system is ready for production of significant batches of specialized microchips.

In the 5-7 year period, GONT capitalization will surpass Bitcoin capitalization.

8.2 Methods of capitalization evaluation

1. Evaluation based on **comparables**

1.1 Comparison with EOS

Now Period 279 - Estimated Market Cap. (for EOS):

$**6,539,000,000** USDT
(279 weeks after EOS ICO)

SWOT
- EOS is growing too fast, without having any real services and conditional “sales”.
- Comparison with EOS, of course, requires correction downwards, because EOS is quite surely promoted by “behind-the-scenes” agreements of mining pools, exchanges, etc.
At the moment, GONT has no arrangements for such a “promotion” of the coin (although they may occur).

- GONT relies on the developing capitalization through the involvement of the maximum number of real services, and without a technical pump of the coin.

1.2 Comparison with IOTA

GONT has considerable groundwork for operating with IoT and, from our point of view, surpasses IOTA technologically.

2. **Evaluation according to the number of transactions of services and the growth rate in the number of services and users of services.**

- GONT system receives a small percentage of transactions through KANT-AlGas.

2.1 Case study of comparable service - **Golos**

We consider Golos to be one of many services that are convenient for implementation on GONT (1000 similar services will operate on GONT in 5-7 years). At the same time, by switching to GONT, Golos could receive a significant lever for development.

The conditional "consumption" of the conditional analog AlGas on Golos nowadays:

"The number of registered users of Golos blockchain exceeded 100 thousand, with more than 30 thousand active participants receiving at least $15,000 per day (more than **$5 million per year**) for the posts. Representatives of the platform informed ForkLog about it."

Using all the advantages of GONT, similar services will increase the "gas turnover" x10 – up to **$50-100 million per year**

*) We think that the activities of Golos users are very convenient for converting in GONT AlGas.

9. Examples of GONT services

9.1 Example of FUPE for HW space

Service:

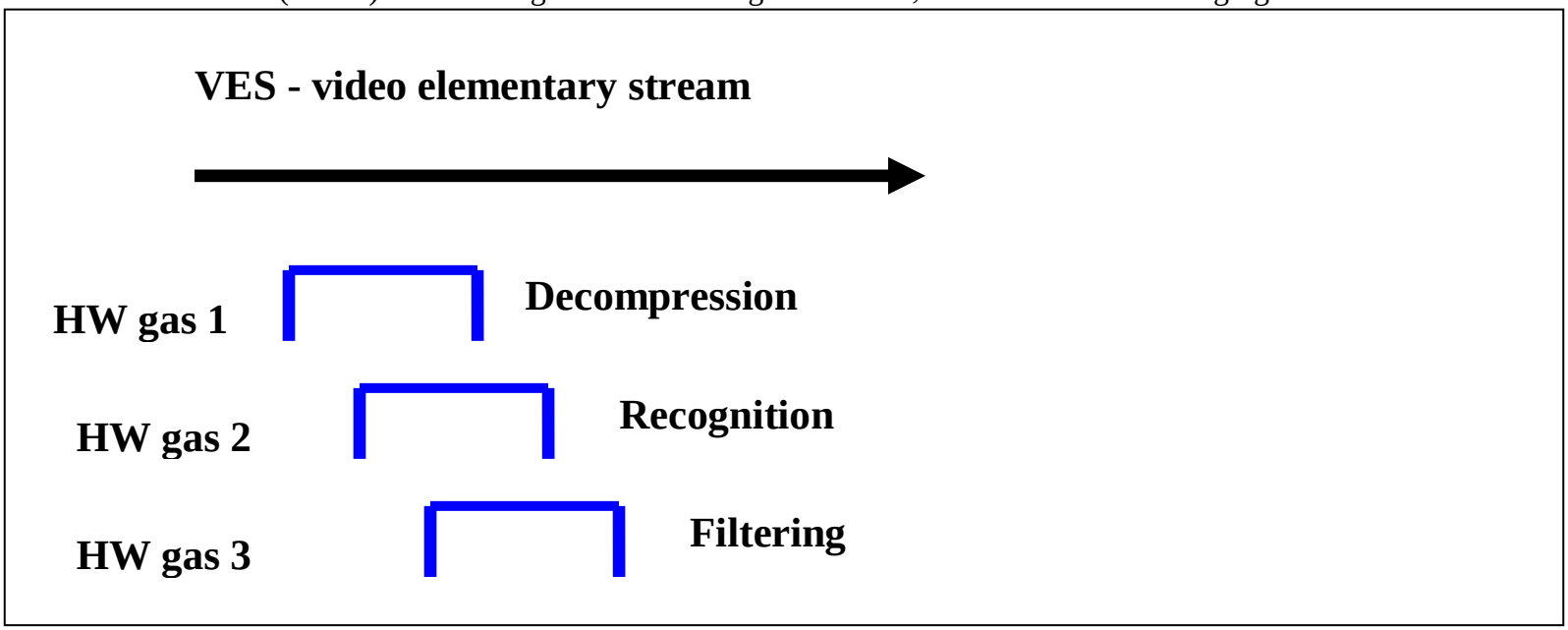
Hardware filter for ads

A service that disables the video stream when ads start on TV. It is a hardware filter for TV ads.

AlGas involved:

1) Decoding the video stream
2) Recognition of images of ads
3) Ad filter

Terms of service:
- various organizations can be the developers of 1), 2), 3)
- fee (TxFee) can be charged for each recognition event, instead of one-off charging.

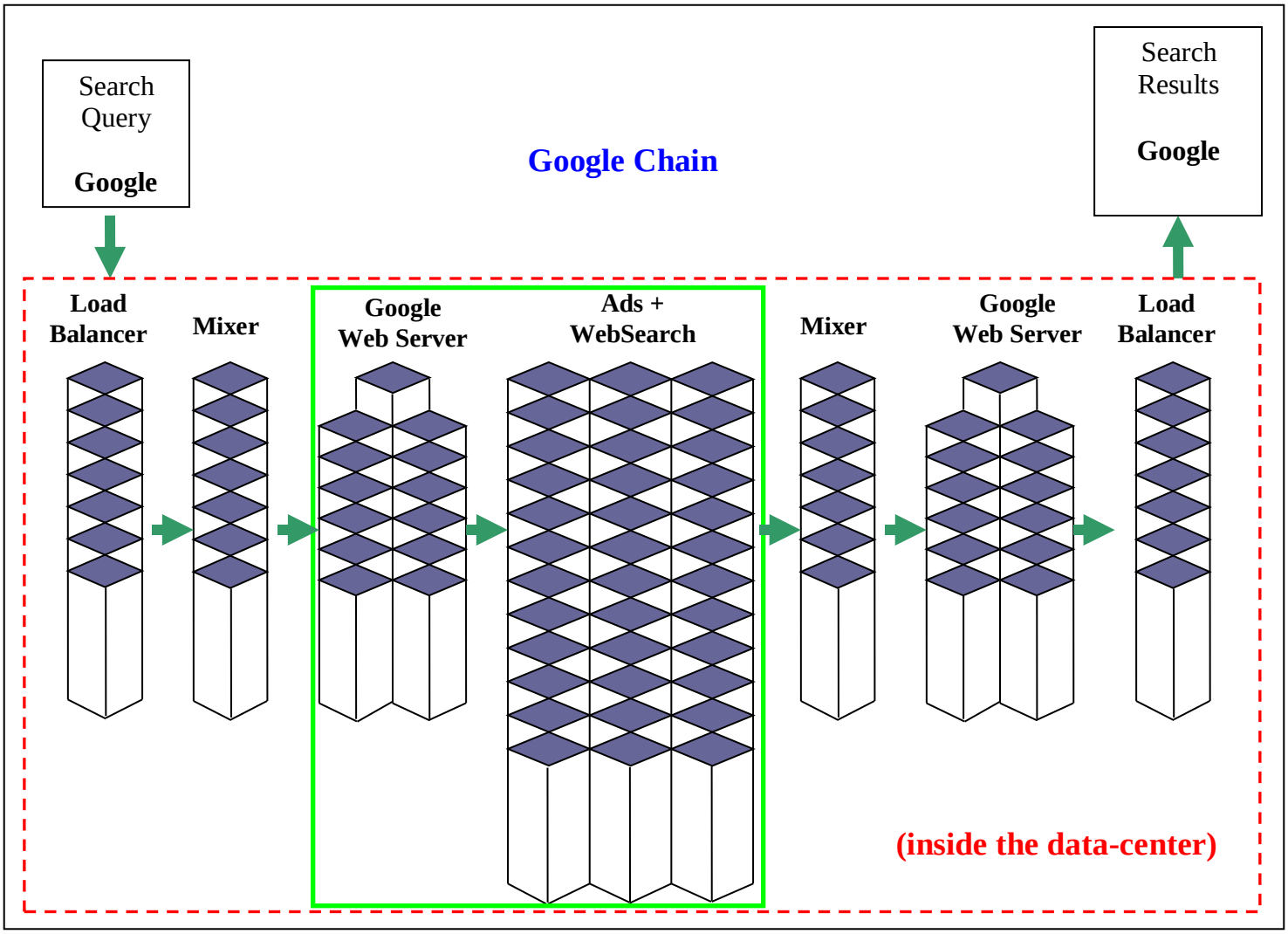


As a result, the end user watches TV without ads.

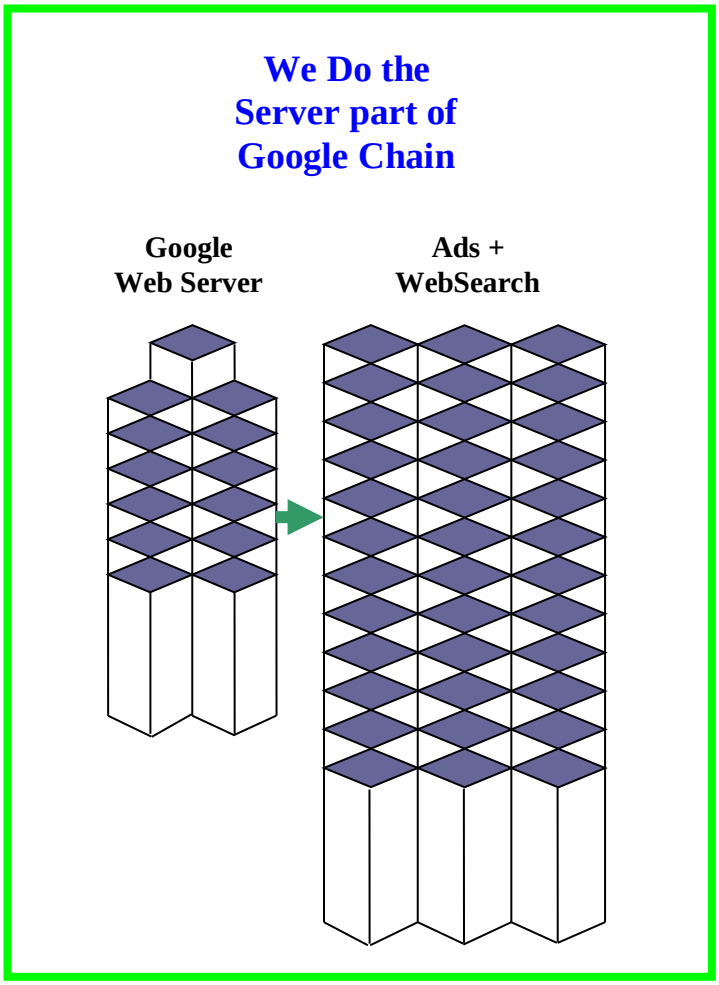
(*) there is already an Apple patent for such a system, but without using blockchain

9.2 What if Google or Yandex operated on GONT

Let’s consider a search query path in Google.

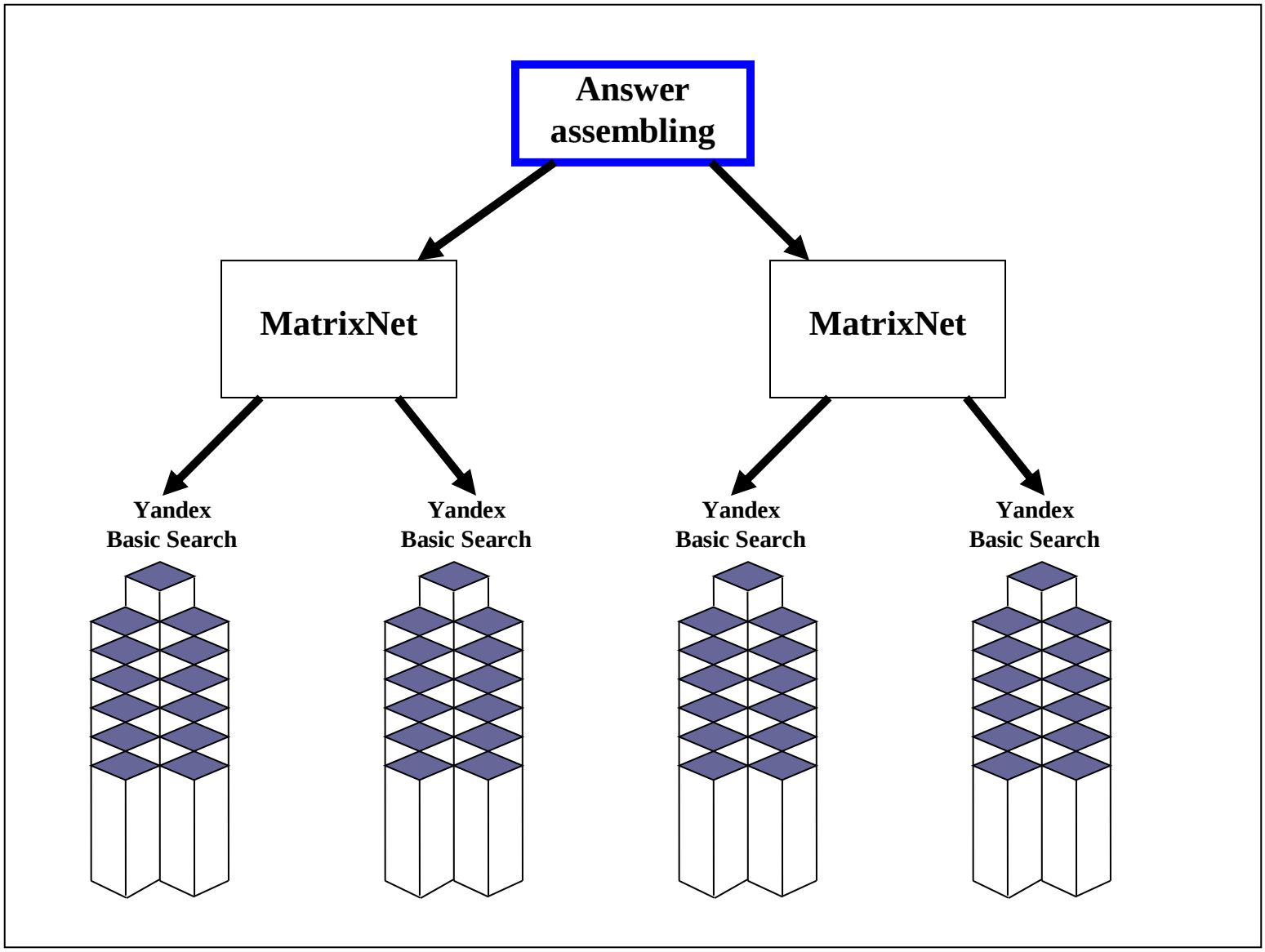


- User's query path on Google servers



- Computational performance of the search query and content ads.

Query path in Yandex. Basic search and MatrixNet



- Simultaneous search on 1000 servers.

Mechanism

- Similar to Blokchain, the basic search in Yandex/Google **is launched immediately on thousands of servers** (indexed servers). And this search is limited in resources (according to the time of the response to the user) – similar to EVM (hard limit of gas).

Then the data, raised from the base search, rise to the **MatrixNet** level for the final ranking with the complex MatrixNet formula developed by computer-aided learning (~ 50,000 lines).

This MatrixNet formula can become an elementary instruction for GONT gas, when displayed on GONT.

But we can also expand the understanding of gas for search.

GONT gas options

1) Basic search instructions. (**G1**)

2) Semantic/MatrixNet search instructions (**G2**)
Ranking according to the ontological data or the formula of computer-aided learning.

3) Instructions for searching and displaying relevant ads (**G3**)

4) Search according to semantics from external suppliers. (**G4**)
- as an example – GONT Tree

GONT Tree provides an opportunity to introduce "private" ontologies of knowledge of companies or knowledge areas to the search.
(For example, ontologies of professional chess players)
Thus, global search becomes deeper (**deep search**).

And these ontologies are not extracted in a trivial way from BigData, available from different sources by default.

Therefore, the gas developer has the motivation to profit off of it, as on his private property.

For example, all people who write books, order knowledge in a certain way. GONT provides additional opportunities for monetization of any books.

Resulting gas: G1 + G2 + G3 + G4

Numerical evaluations

One query to Google costs ~ $0.01 (this **energy is enough to boil the teapot**). It means that the gas consumption has a completely visual and tangible equivalent.
Obviously, in this case it is possible to build a "gas" pattern on blockchain.

The number of search factors in **MatrixNet > 1000** (the number of GONT gas instructions).
The number of search factors from private ONTO for this user query can be ~ 1000 as well.