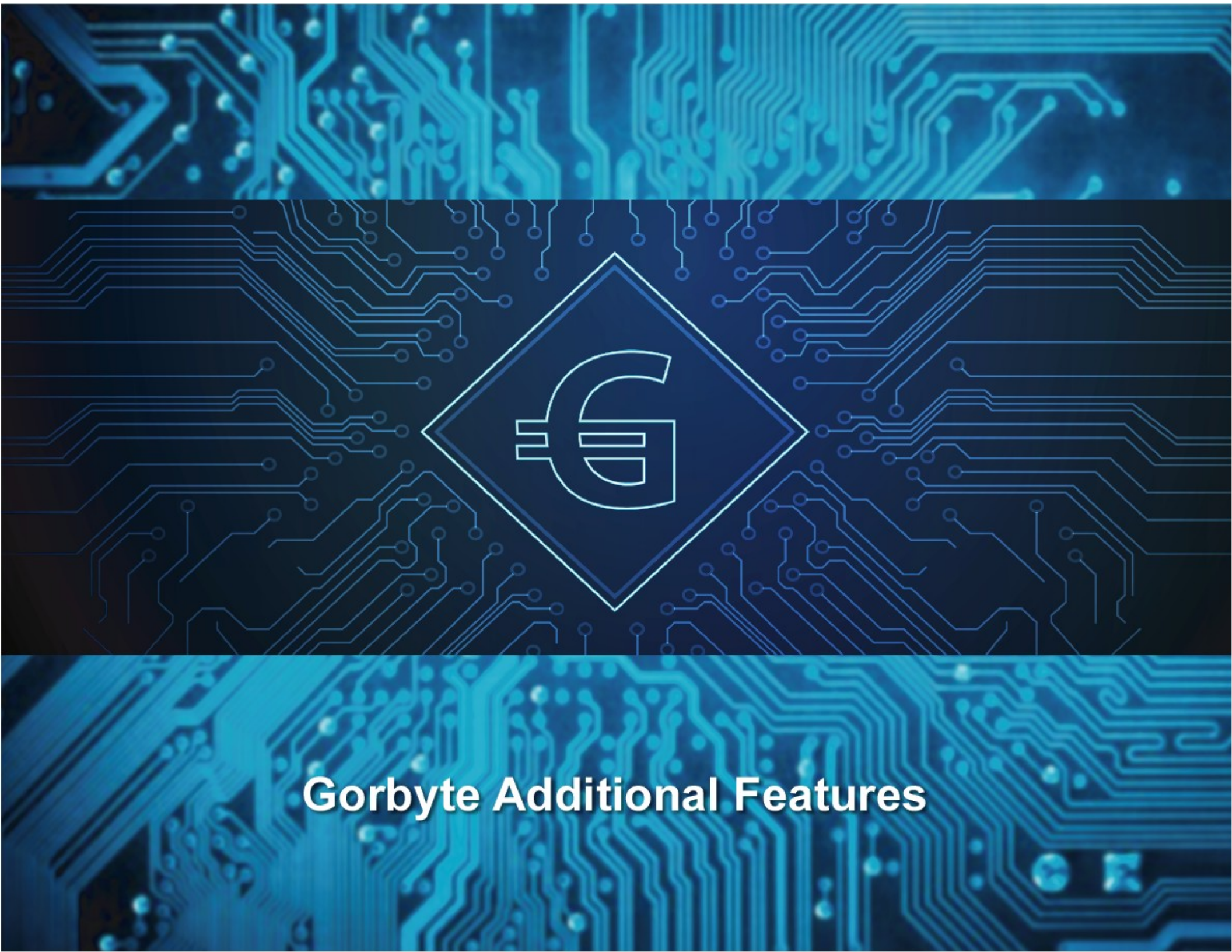

Gorbyte Additional Features

Table of Contents

1. Gorbyte Additional Features

The basic concept of the Gorbyte crypto-network and its differences with current crypto-networks were described in the Gorbyte Introduction document.

This document describes the additional features and functionality that have been added to the Gorbyte design.

These additional features fill the gaps in the functionality that is needed for the operation and governance of a modern digital currency.

Please note that our reference for this additional functionality is Bitcoin. Some of the features introduced here can be programmed or are already included in other crypto-networks. For example, secure vaults can be implemented using smart contracts in Ethereum. However neither Ethereum nor other unpermissioned crypto-networks support general distributed applications (GApps) using the blockchain, but running off the blockchain.

Additionally, we are not aware of any digital currency that includes elasticity of its money supply, or redistributes its earnings as dividends to currency holders.

A glossary of terms and abbreviations used in this document is provided in Appendix C.

2. Additional Functionality Overview

The Gorbyte Introduction document includes an explanation of how Gorbyte maintains most of the Bitcoin functionality with respect to transaction encryption and verification, but replaces its consensus mechanism with a distributed majority agreement process.

The introduction also explains how a virtual BRUD device can prevent identity proliferation. The definition of BRUD devices is provided in “The BRUD Architecture” document.

In addition, Gorbyte introduces further functionality, when compared to other crypto-networks (See next sub-section).

2.1 Additional Functionality Comparison

The following table compares the additional functionality of Gorbyte with current Bitcoin and Ethereum functionality:

Gorbyte	Bitcoin	Ethereum
Support for **Secure Vaults**, to improve the security of user holdings. *See section 3.*	Covenants and Vaults could be added.	Vaults can be provided through smart contracts.
Achieves **elasticity of the money supply**, and more stable prices. *See section 4.3.*	No such functionality.	No such functionality.
Distributes the proceeds from GApps among currency holders in proportion to their stake. *See section 4.4.*	No such profit sharing.	No such profit sharing.
Will allow stakeholders, in proportion to their stake, to **introduce a target artificial inflation rate**. *See section 4.5.*	No such functionality.	No such functionality.
Will allow stakeholders, in proportion to their stake, **to control the currency re-basing index**. *See Appendix B.*	No such functionality.	No such functionality.

Gorbyte	Bitcoin	Ethereum
Will allow stakeholders, in proportion to their stake, **the means to exercise governance** on the crypto-network through **polls**. *See section 5.*	No such functionality.	No such functionality.
Ability to easily **replace its encryption methods**, when necessary. *See section 6.*	Fixed encryption method. A replacement may be possible.	Fixed encryption method. Replacement are possible through forks.
Support for **peer-to-peer distributed applications**. *See section 7.*	Limited script language.	Smart contracts are costly and limited.
Allows for the unique **identification** of nodes and users **through** BRUD devices and biometric technology. *See "The BRUD Architecture".*	No such functionality.	No such functionality.

Note that Ethereum implements a language allowing for a limited set of distributed applications running on the blockchain (Smart Contracts). However, this language is used also for tokens and complex financial contracts, opening the opportunity for user mistakes and theft.

Gorbyte will use simple scripts for financial transactions, and will use objects methods specific to each distributed application framework for its GApps.

3. Vaults

3.1 Covenants and Vaults

With the tamper-proof ***BRDG*** hardware device, user private keys will be protected by hardware means and their funds will be secure from theft.

In its first implementation, when using virtual BRUD devices, Gorbyte will implement Covenants and Vaults[1] to better secure users' funds from theft.

However, an extension (Secure Vaults, see section 3.2) will be introduced to prevent the loss of funds even when the primary key and the recovery key have both been compromised.

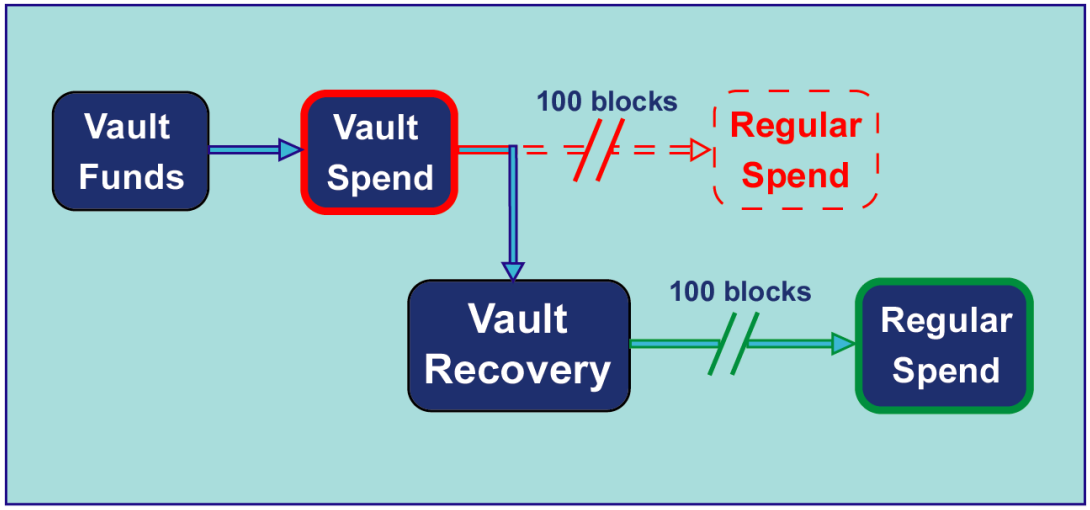
3.1.1 The Objective of Vaults

> *"Bitcoin funds are, by and large, protected by sets of cryptographic secret keys. Whoever knows those keys can instantly, anonymously, and irrevocably move the funds by spending the transaction outputs in which they are represented. This makes Bitcoin private key theft an attractive target for thieves"* (Ibid. Ref. 1).

Vaults remove the incentive for malicious attackers to steal the encryption keys by preventing the attacker from gaining access to stolen funds.

3.1.2 Vaults

Vaults prevent an attacker from instantly moving funds from a victim's wallet by enforcing a delay for the transfer of those funds.



Drawing 1: Vaults

In the above diagram, an attacker's spending attempt (red) is interrupted by a vault recovery issued by the legitimate owner, followed by a regular spend of the legitimate owner (green).

Funds associated with a vault transaction cannot be released immediately. The key idea of vaults is that the spending transaction must be placed publicly on the blockchain, with its output locked for a specified amount of time.

During this period, the owner of the funds can abort the release of the funds using a recovery key (preferably placed in cold storage) to send the funds back to himself with a new spending transaction, thereby denying the payout to the attacker.

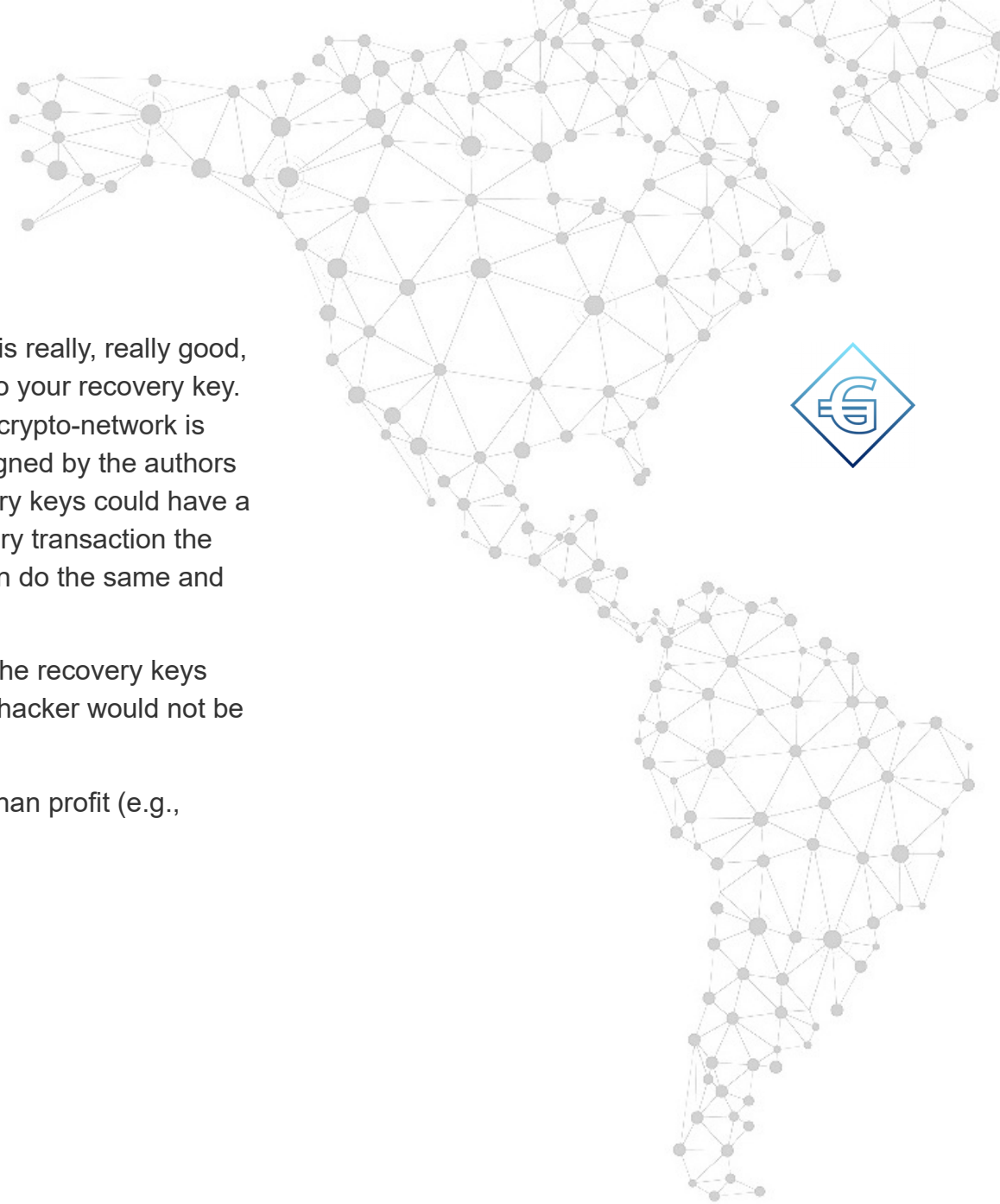
3.1.3 Support of Vaults

This sub-section has been adapted from articles published by the authors of the document in Ref. [1].

Vaults are the decentralized version of you calling your bank to report a stolen credit card -- it renders the attacker's transactions null and void. And here's the interesting part: in so doing, vaults demotivate key theft in the first place. An attacker who knows that he will not be able to get away with theft is less likely to attack in the first place, compared to current Bitcoin attackers who are guaranteed that their hacking efforts will be handsomely rewarded.

You send your money to a vault address that you yourself create. Every vault address has a vault key and a recovery key. When spending money from the vault address with the corresponding vault key, you must wait for a predefined amount of time (called the unvaulting period) that you established at the time you created the vault - say, 24 hours.

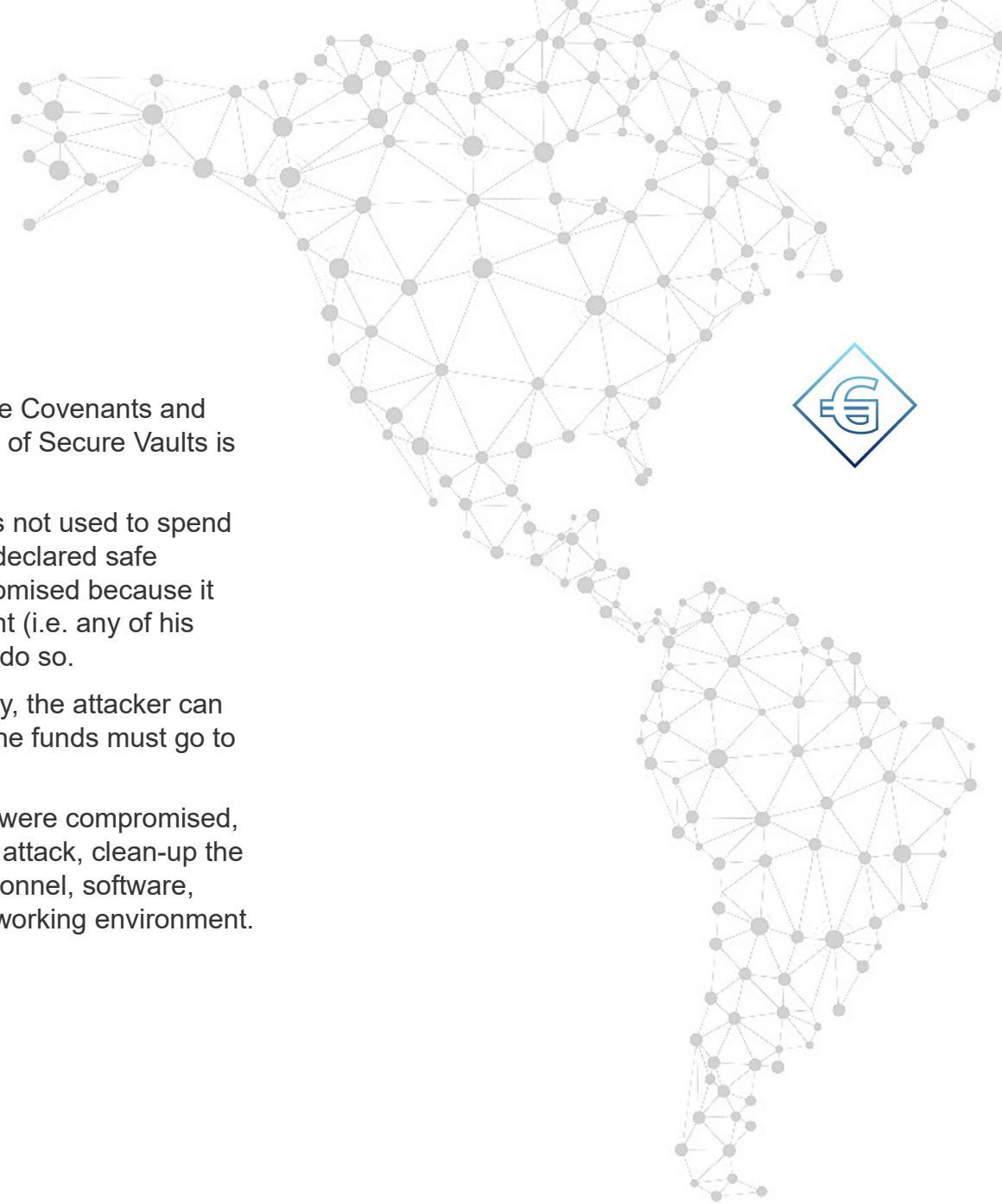
When all goes well, your vault funds are unlocked after the unvaulting period and you can move them to a standard address and subsequently spend them in the usual way. Now, in case a Hacker gets a hold of your vault key, you have 24 hours to revert any transaction issued by the hacker, using the recovery key. His theft, essentially, gets undone, and the funds are diverted unilaterally to their rightful owner. It's like an "undo" facility that the modern banking world relies on.



Now, the reader will ask what happens when the hacker is really, really good, and he lies in wait to steal not just your vault key, but also your recovery key. That is, he has thoroughly cloned you and, as far as the crypto-network is concerned, is indistinguishable from you. Vaults, as designed by the authors in Ref. [1], devised a protection for this case. The recovery keys could have a similar lock period, allowing you to perpetually revert every transaction the hacker makes. Unfortunately, at this point, the hacker can do the same and revert every transaction you make.

To avoid a perpetual standoff, the authors point out that the recovery keys can also burn the funds, so no one gets the money. The hacker would not be able to collect the proceeds from his theft.

Hackers however are often motivated by reasons other than profit (e.g., disruption) when engaging in cyber-attacks.



3.2 Secure Vaults

Gorbyte implements Secure Vaults as an extension of the Covenants and Vaults as specified in the previous section. The objective of Secure Vaults is to drastically reduce the risk of losing funds.

The idea behind Secure Vaults is that the recovery key is not used to spend the funds, but the funds are always sent to a previously declared safe address. The safe address private key cannot be compromised because it has never entered the compromised working environment (i.e. any of his devices connected to the internet) and does not need to do so.

If the attacker has also gained access to the recovery key, the attacker can only use it to enforce the covenant, which requires that the funds must go to the safe address.

This procedure, used in the rare cases where both keys were compromised, gives the owner the time to understand the nature of the attack, clean-up the compromised environment, change his procedures, personnel, software, physical keys, etc. before moving his funds again to his working environment.

More implementation details are provided in Appendix A.

4. Currency matters

This section describes Gorbyte's total reserve, its basic currency, its re-based currency and the mechanism for distributing the proceeds from network fees to currency holders.

4.1 Comparison

The following table compares Gorbyte, Bitcoin and Ethereum on currency matters.

As we have seen in the comparison table in section 2.1, Gorbyte will introduce the following unique functionality with relation to currency:

- the mechanisms to achieve elasticity and price stability, and a method to control the rebasing index (See appendix B);
- a mechanism for profit sharing of the fees paid to the network for GApps in the Distributed Operating Environment (DOE); and
- a mechanism for introducing a target artificial inflation rate.

The following table compares additional issues related to currency:

Gorbyte	Bitcoin	Ethereum
Will initially create twice the initial amount raised, converted to basic gors. There is **no functionality to create new currency**, hence less risk of theft, and no intrinsic devaluation.	Miners will create 21,000,000 bitcoins. New currency creation is possible and ongoing.	There is no maximum. More coins are created every year. New currency creation is possible and ongoing.
No fees or rewards need to be paid to miners or validators. Thus, there is **no intrinsic currency devaluation**.	The currency has an intrinsic devaluation factor.	The currency has an intrinsic devaluation factor. This may be reduced if PoS is implemented.

4.2 Gorbyte Total Reserve

The total reserve amount was established by the Gorbyte Foundation as double the amount in BTC or ETH, pledged during the funding period, expressed in basegors (See Appendix C for the definitions of “basegors” and “Gors”.
See also the Gorbyte Foundation web site, “Campaign Terms” document).

The total amount of Gorbyte's money supply is expressed in basegors. This is established when its initial total reserve is created in the blockchain genesis block. In Gorbyte, no mechanism exists to create new basegors.

Currency nominal values in transactions are expressed in basegors.

The total amount of unspent outputs of addresses that are part of a user wallet can be calculated from the blockchain record. This is expressed in basegors. basegors can be easily converted to Gors and vice-versa by using a conversion function described in the Gorbyte Specifications document.

However, basegors will constantly acquire value due to the fees paid for general distributed application services (GApps). This acquired value is independent from fluctuations due to currency demand. Both appreciation and fluctuation effects are resolved through the mechanism of elasticity, introduced in the next section.

4.3 Elasticity of the Money Supply

The concepts of elasticity and price stability have been introduced in Gorbyte thanks to the research work of Dr. Ferdinando M. Ametrano[2].

While no new basic currency is ever created, its rebased Ğors (what users see, own and deal with) will steadily and automatically increase by the amounts necessary to counteract increased demand, and take care of the tendency towards an increasing currency value.

4.3.1 Rationale

The rationale for an elastic money supply, is to provide:

1. a reduction (smoothing) of currency fluctuations, and
2. stable prices when the purchasing power of the currency could change considerably with increased demand.

The **first objective** is to *prevent those conditions that discourage potential users* of the currency.

The **second objective** is to *encourage the acceptance of the currency* for:

- contracts, such as salaries, rental agreements, mortgages, etc.;
- currency uses where consumers may be weary of an unstable digital currency, such as pensions, benefits, savings, etc.; and
- financial estimates in the areas of forecasting, setting consumer prices, estimating costs, comparing suppliers from different countries, appraisals, etc.

These objectives will be accomplished without the need of a central monetary authority.

The purchasing power of the rebased currency unit (~~G~~ors) can be maintained relatively stable. This can be accomplished through elasticity: the ability to increase the amount of ~~G~~ors in peoples' wallets in reaction to currency market forces causing growing demand.

4.4 An Extra Incentive for Currency Owners

The Gorbyte crypto-network will not be corporately owned or controlled. In addition, there are no fees or rewards going to miners. The proceeds from fees charged for running general distributed processing applications (GApps) are shared among all currency owners, in proportion to their holdings.

Because of its currency elasticity, the rate of increase in basic currency value can be counteracted and smoothed out. Currency holders will see their ~~*Gors*~~ occasionally increase in number, by variable amounts.

This automatic mechanism encourages and rewards currency ownership. The Gorbyte fees for heavy load GApps are handled very simply at the transaction level. These fees do not go to miners, as in PoW networks, or to validators, as in PoS networks, but to Gorbyte stakeholders (currency owners).

This is achieved by:

- Removing from circulation (or burning) the amount in basegors paid in fees to the Gorbyte crypto-network (the opposite of creating money).
- Allowing the process of elasticity of the money supply, described in Section 4.3, to work. When users will see the amount of ~~*Gors*~~ in their wallets, as presented to them by exchanges and wallet applications, they will find these amounts slightly increased.

The consequent trend of increase in wallets' worth (i.e.: currency value multiplied by the quantity of Ⅎors owned), benefits everyone holding Gorbyte currency in proportion to the fees paid to the network.

With this mechanism, holding Gorbyte currency becomes an investment in Gorbyte futures and its ability to develop useful distributed applications.

This will be a particularly attractive form of high liquidity investment, as Gorbyte's wallets will, by design, steadily increase in worth.

4.5 Introducing a Target Artificial Inflation Rate

With the introduction of elasticity, we have reduced price volatility, and introduced variability of the amount of rebased currency in users' wallets. With this mechanism at its disposal, Gorbyte can allow stakeholders to control some of the currency parameters, such as the target rebasing index (introduced in Appendix B)and a target artificial inflation rate, explained below.

A yearly target artificial inflation rate produces an equivalent decrease in the value of rebased ~~G~~ors, compensated by an increase in the number of ~~G~~ors in people's wallets. This target rate can be changed by consensus agreement.

In fiat currencies such distribution of new printed money during inflationary periods does not happen. Peoples' currency (in cash or bank accounts) becomes less valuable, but no compensation is provided by the banks.

An artificial inflation of the currency is used to counteract:

- a period of higher currency demand, and
- the reduced money supply in basegors, corresponding to the fees paid for GApps, as explained in the previous sub-section.

Conversely, an increase in currency value would occur when the demand for the currency increases and the currency value is below the target inflation rate.

Psychological factors

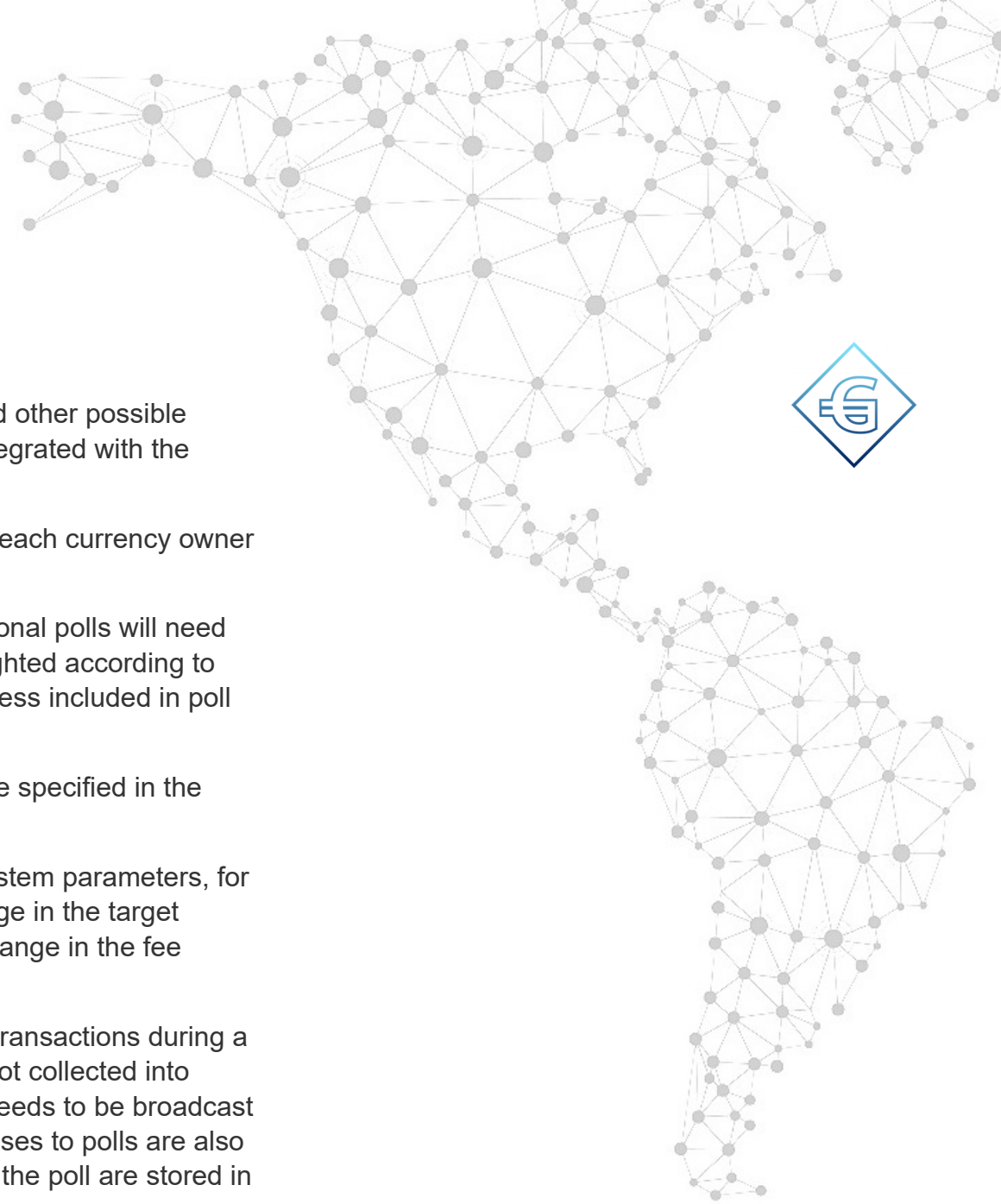
When an automatic inflation is triggered, counteracting a period of higher currency demand, the surprise of finding more money than expected in a wallet is certainly not traumatic for the wallet owner. However, if a person opened his wallet to pay a bill and found less money than expected, this would indeed be a surprise. For this reason, Gorbyte's elasticity allows for increases in the quantity of Gors in people's wallets, but allows for the value to drop when there is a drop in demand.

A slight inflation rate has another psychological advantage: it attracts savings and investments, because people are more likely to appreciate a growing number of Gors, rather than a higher purchasing power of their Gors.

One more reasons to maintain a small (positive) inflation rate is delayed payments: With fiat currencies people have the option, when paying an invoice, to delay their payment. Often, they do so while waiting for a payment from their own customers or for a salary installment. Because the inflation rate is normally above zero, customers know that they will not lose money by delaying a payment. Suppliers provide their customers with this flexibility (e.g.: a 30-day invoice).

With a negative inflation rate, customers would lose the flexibility of delayed payments, because the agreed invoice amount would not change, while the currency appreciated in value.

There are other considerations that generally favor a policy of maintaining interest rates low, but above zero. Such questions of monetary policy are beyond the scope of this document.



5. Polls for Governance Decisions

The Gorbyte network handles version changes, polls and other possible cooperative or Governance decisions in a way that is integrated with the blockchain.

Polls are weighted according to the stake percentage of each currency owner who cares to respond to the poll.

Any user can initiate an occasional poll for a fee. Occasional polls will need to be advertised by the proponents. These polls are weighted according to the stake (in unspent basegors) associated to each address included in poll responses.

The poll mechanism, poll types and message formats are specified in the Gorbyte Specifications.

Occasional polls can be used to propose a change to system parameters, for network governance. For example, for proposing a change in the target yearly inflation rate of the currency, or for proposing a change in the fee structure.

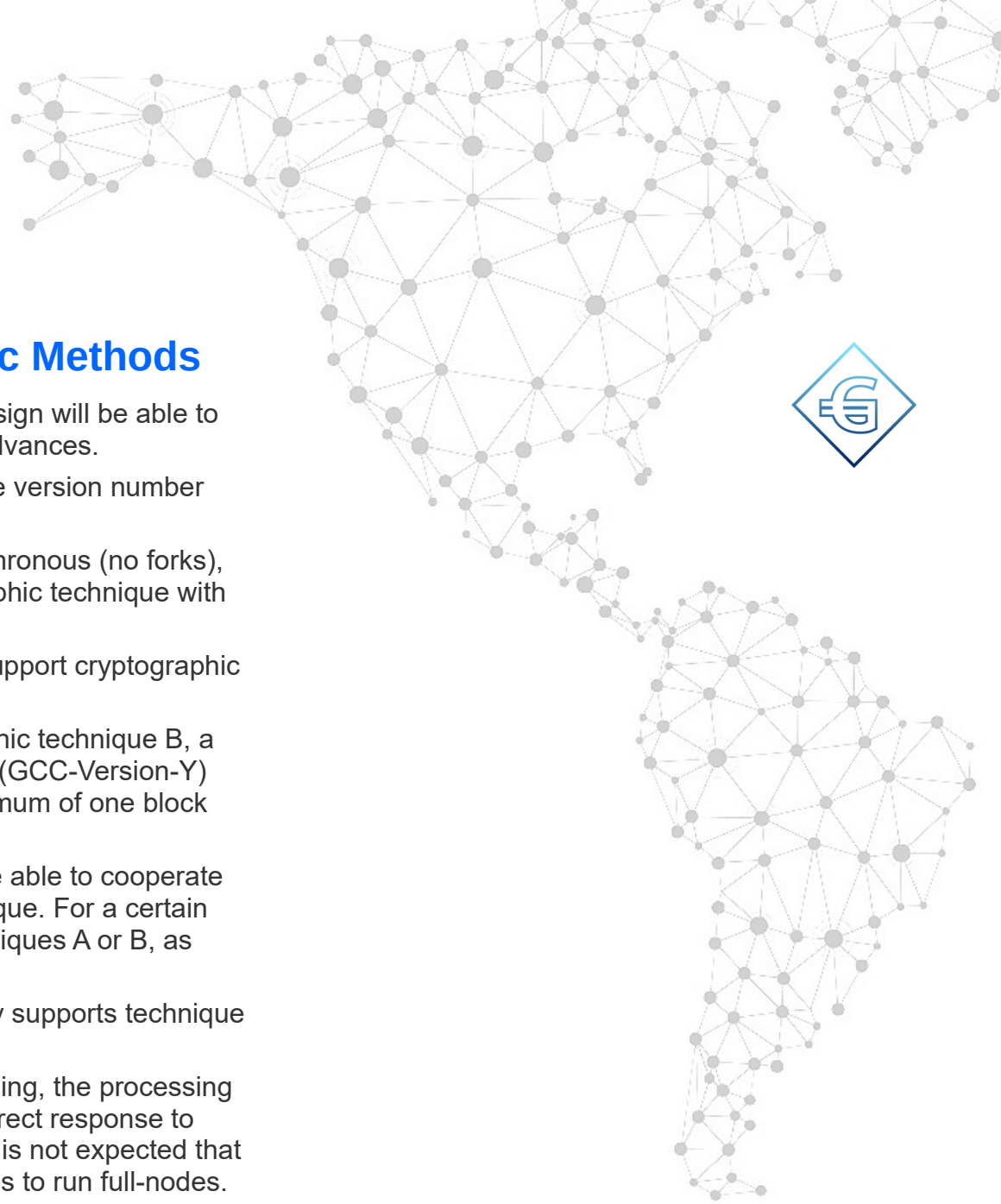
Poll requests are broadcast in the same way as regular transactions during a summary period (e.g., approximately one day), but are not collected into regular blocks. They prompt nodes for a response that needs to be broadcast before the end of the same summary period. The responses to polls are also broadcast as regular transactions. The agreed results of the poll are stored in special blocks, called Summary Blocks.

The poll type used by nodes to choose the target rebasing index is automatic. It is implicit at the beginning of each summary period (i.e., a poll request is not needed). Thus, every node just needs to issue a poll response for that poll type before the end of the summary period. Summary periods are the periods in between Summary Blocks.

Summary Blocks are those with a block number ending with a number of zero bits, that determines the period's length (for example, 10 zero bits, for 1024 blocks). The Summary Blocks are used for summary data used to speed-up calculations. For example, the calculations of unspent balances in basegors associated to addresses.

Summary Block headers are also used to contain special fields related to periodic activities and for polls. One of these fields is the target rebasing index field. This field is the result of the agreement reached through the automatic target rebasing index poll agreed by all active nodes about once a day.

The rebasing mechanism, rebasing fields and conversion function are explained more fully in the Gorbyte Specifications.



6. Ability to Upgrade Cryptographic Methods

In addition to normal software upgrades, the Gorbyte design will be able to upgrade to new cryptographic methods as technology advances.

Transaction headers include two fields which indicate the version number and the type of cryptographic technique used.

Because the addition of blocks to the blockchain is synchronous (no forks), nodes can upgrade to a new version or a new cryptographic technique with minimum disruption.

For example, the Gorbyte Client: GCC-Version-X may support cryptographic technique A.

When an upgrade is required, for example to cryptographic technique B, a node may download a new version of the Gorbyte client (GCC-Version-Y) and re-synchronize. This will disrupt this node for a minimum of one block period and a maximum of a few block periods.

However, after an upgraded node restarts, it must still be able to cooperate with peer nodes using the previous cryptographic technique. For a certain period, the new GCC-Version-Y must support both techniques A or B, as indicated in the transaction header.

Eventually a GCC-Version-Z may be introduced that only supports technique B.

As cryptographic techniques may become more demanding, the processing power required of nodes will also increase. Both are a direct response to progress in computing power, and related costs. Thus, it is not expected that improvements in cryptography will require special devices to run full-nodes.

7. The Distributed Operating Environment

The Gorbyte design will include support for general distributed applications (GApps). These can use the blockchain, but can do most of their processing and data storage off the blockchain. GApps represent a much broader set of distributed applications than the DApps pioneered by Ethereum.
The Gorbyte design will be developed according to a Distributed Operating Environment model, which will allow different levels of flexibility for different Application Frameworks.

Basic financial transactions and critical contracts will use a script language, while smart contracts may be used for less critical applications.

More complex distributed applications will use registration objects on the blockchain to identify and verify the peer participants (the service providers and their users). The bulk of the interaction among the peer players, the data exchanges and data storage will happen off-the-blockchain.

This approach is extremely scalable and will allow the execution of more complex distributed applications.

The Distributed Operating Environment (DOE), will gain Gorbyte Inc. a foremost position in the industry of blockchain-based general distributed applications (GApps)[3].

APPENDICES

A Implementation of Secure Vaults

Gorbyte implements the Secure Vaults an extension of Covenants and Vaults described in Ref.[1]. The following describes the problem and its resolution.

Problem

If a hacker can steal a private key, most likely he/she has acquired access to the funds owner's private environment, which is therefore compromised. Possibly the recovery private key has also been stolen.

Transferring the recovered funds back to the owner, if done within a limited time frame, may cause another security breach. If the owner needs to recover the funds in a hurry, he/she may have no alternative but to use the same compromised environment.

Objective

Gorbyte's risk-free transaction extension avoids the tit-for-tat case where the owner can prevent the hacker from collecting the stolen funds only by burning his funds.

Implementation

The Recovery key (specified in the document in Ref. [1]) is now split into two: a Stop Payment key, and a Recuperation key. The first needs to be handy, probably to be used in the compromised working environment, to stop a payment. The second needs to be kept in a safe environment.

To set up a Risk-free transaction vault, the following are the key pairs needed:

1. The ***Vault key*** (as specified in the document in Ref. [1]) is the key pair used for “Secure Vaults”;
2. The ***Stop Payment key***, a key pair used only for stopping payments; and
3. The ***Recuperation key***, a key pair used at the end of the recovery and cleanup process to restore the funds to the original owner.

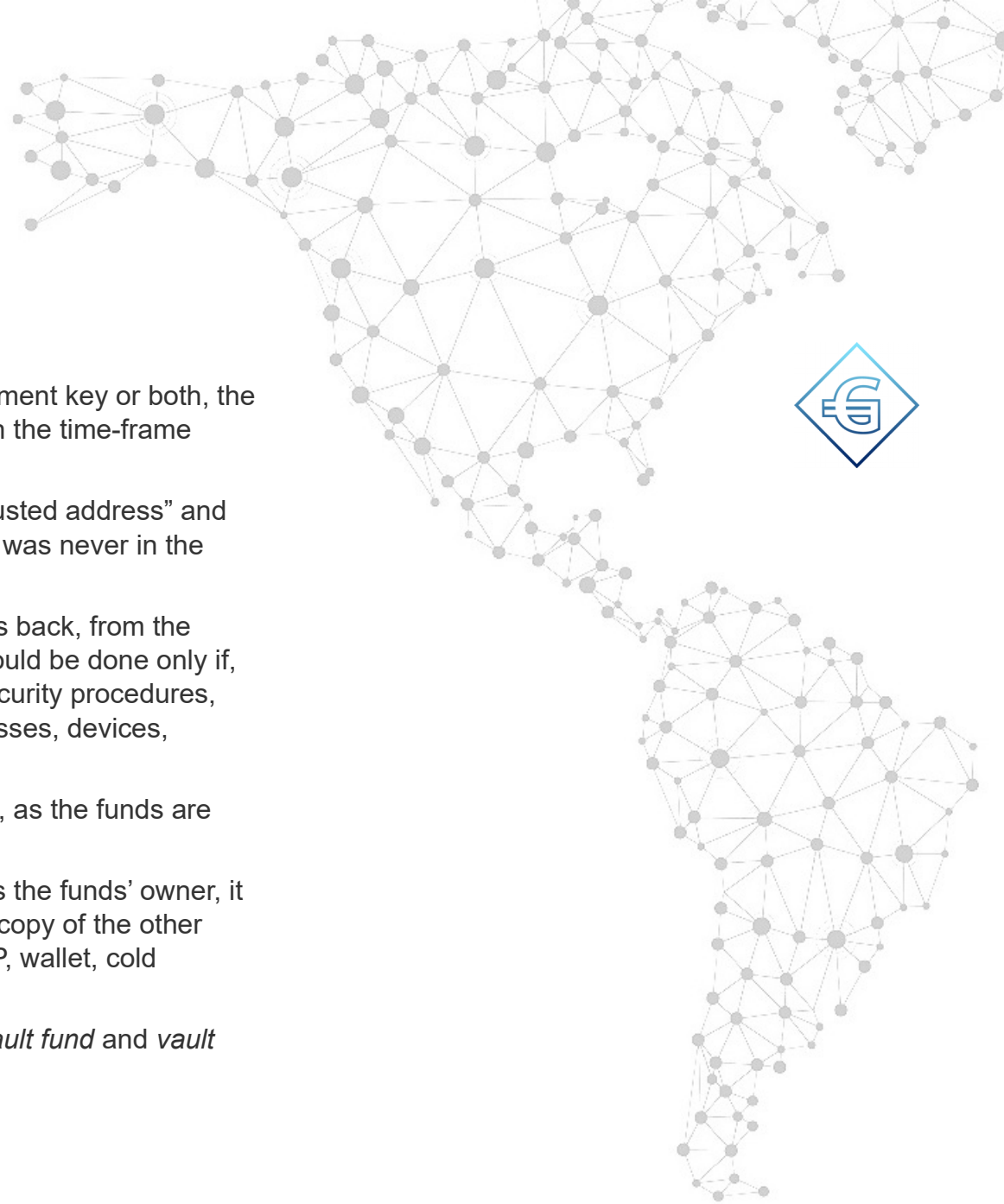
The Recuperation trusted address can possibly be provided by a trusted third party. It should be impossible to reach the private Recuperation key address from the owner’s working environment through electronic communication or other insecure physical means.

As per the specifications of vault, the owner uses its Vault Key to make delayed payments.

In case a theft occurs, the owner can interrupt his risk-free transaction payment from his vault by initiating the Stop Payment process.

This uses the Stop Payment key ONLY for signature identification (i.e., only the owner of the Stop Payment key can stop payments from his vault). Ownership of the Stop Payment key, shall not imply ownership of the funds.

The Stop Payment process always sends the funds to the “trusted address” associated with the Recuperation key.



Result

Whether the hacker steals the Vault key or the Stop Payment key or both, the owner can stop any payment issued from the vault within the time-frame chosen for the vault (the unvaulting period).

After a Stop Payment, the funds will always go to the “trusted address” and will not be lost, because the key of the “trusted address” was never in the compromised working environment.

Finally, after a hacker attack, the transferring of the funds back, from the trusted address to the owner’s working environment, should be done only if, and when, he/she has reviewed and re-organized his security procedures, keys, ISP security, network connection, secretary, addresses, devices, passwords, door locks, etc.

These actions can now be done without time constraints, as the funds are now out of the vault, but safe.

NOTE: If the third party needs to be the same person as the funds’ owner, it is suggested to ensure that the Recuperation key and a copy of the other keys, are kept in a different, safe environment (device, IP, wallet, cold storage, office, people, etc.)

The above logic will be formalized in the scripts of the *vault fund* and *vault spend* transactions.

B Issues related to currency

Seigniorage: Distribution of New Money

With current fiat currencies, when the central banks increase the money supply (i.e.: print new money), this is not distributed to all people (currency holders) at the same time. The new money, available before a price correction happens, is first handled by the banks, then by the people receiving money directly from government (government employees and contractors). These people start using the new money in the open market, as their money is still exchanged at the market value before the inflation. However, this introduction of new money soon determines a devaluation. Finally, the money, now devalued, enters general circulation.

Regular people and businesses do not get a share of the new money in their bank accounts to compensate for the devaluation. On the contrary, their money is devalued – it does not buy what it could buy the previous month.

With Gorbyte, the seigniorage and inflation revenues that have been channeled through retail banks to sustain public debt are going to subside.

Elasticity

The automatic mechanism of elasticity introduced in Gorbyte guaranties that the variations in currency demand and the paid dividends will affect all currency holders in proportion to their stake.

To keep the value of Gors as stable as possible, currency high demand (which would imply a value appreciation) has to be counteracted by an increase of the money supply.

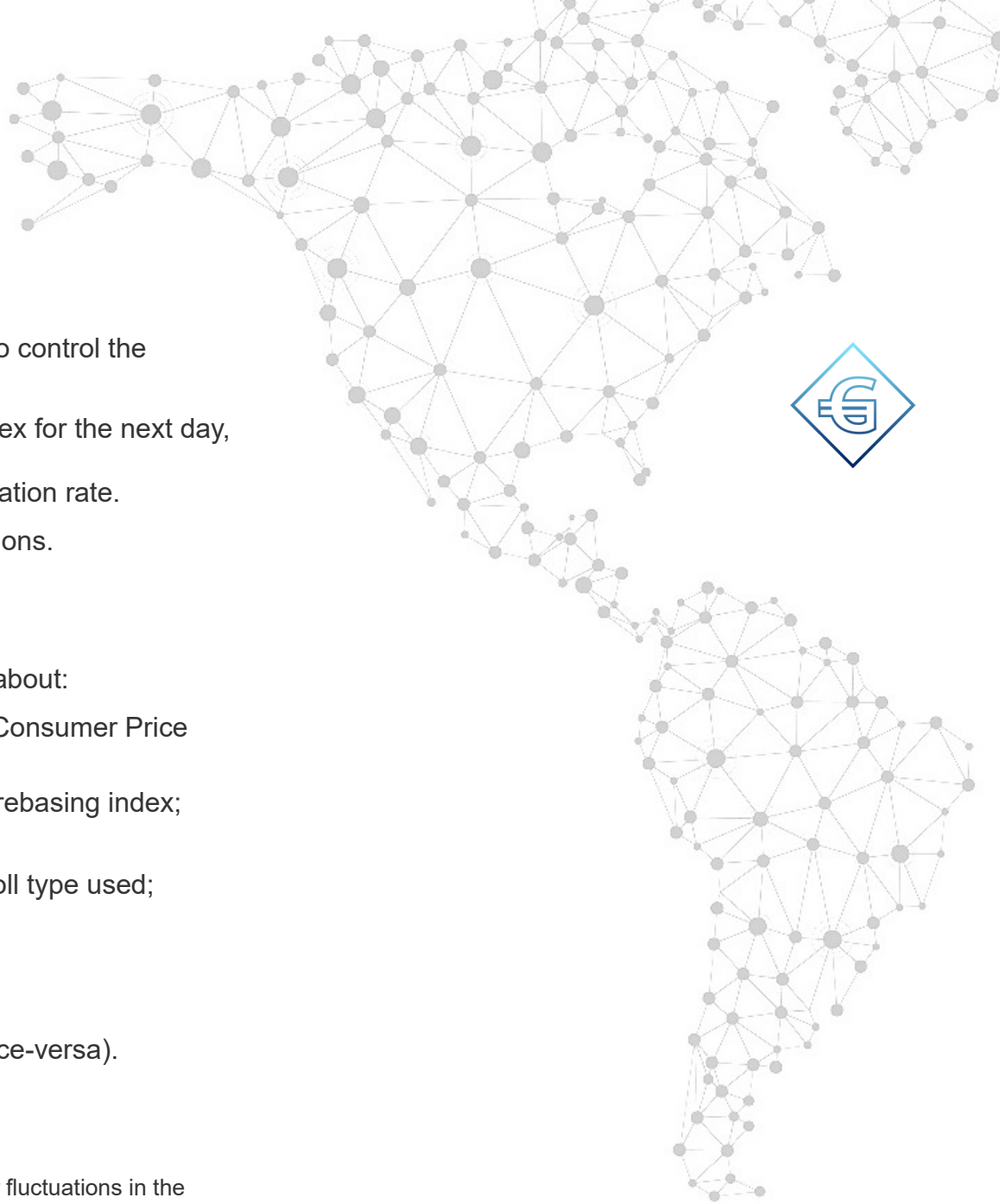
People holding Gorbyte currency will see their nominal amount of rebased currency (Gors) increase, proportionally to the inflating corrections introduced to compensate for the increased demand. However, if demand dropped, the value of rebased Gors is allowed to drop. As demand grows again, the value is allowed to increase, but only up to a target value line; then, if demand continues to increase, rebasing kicks in again and new Gors are created.

In addition, since the rebased Gors are calculated as a multiplier of basegors, the appreciation of basegors (due to GApp fees paid to the network) implies an appreciation of rebased Gors.

In Gorbyte, these inflating corrections (new money) are determined by all users of the network by an automatic daily poll mechanism which takes into account people's stake in the currency. This poll mechanism is introduced in section 5.

Money Volatility

Referring again to Dr. Ametrano's document in Ref.[2], we can say that money volatility is an intrinsic property of the dynamic nature of demand. The variations of demand over time cannot be artificially governed. The resulting change in value cannot be stopped, but can be statistically smoothed.



The two polling mechanisms Gorbyte stakeholders use to control the currency value[A] are:

- a daily automatic poll setting a target rebasing index for the next day, and
- occasional polls to change the yearly inflation/deflation rate.

Both mechanisms intend to avoid high currency fluctuations.

The polling mechanism is introduced in section 5.

More details

The “Gorbyte Specifications” document includes details about:

- The relation between the rebasing index and the Consumer Price Index;
- The alternatives available for choosing the target rebasing index;
- How user nodes propose a target rebasing index;
- The range of the rebasing index choice and the poll type used;
- How the agreed rebasing index is applied;
- The calculation of the rebasing index;
- How the target inflation rate is agreed; and
- The conversion function (basegors to ~~G~~ors and vice-versa).

(A) I.e., whether to allow fluctuations in the **value** of the currency, or fluctuations in the **number** of ~~G~~ors in their wallets, or a combination of both.

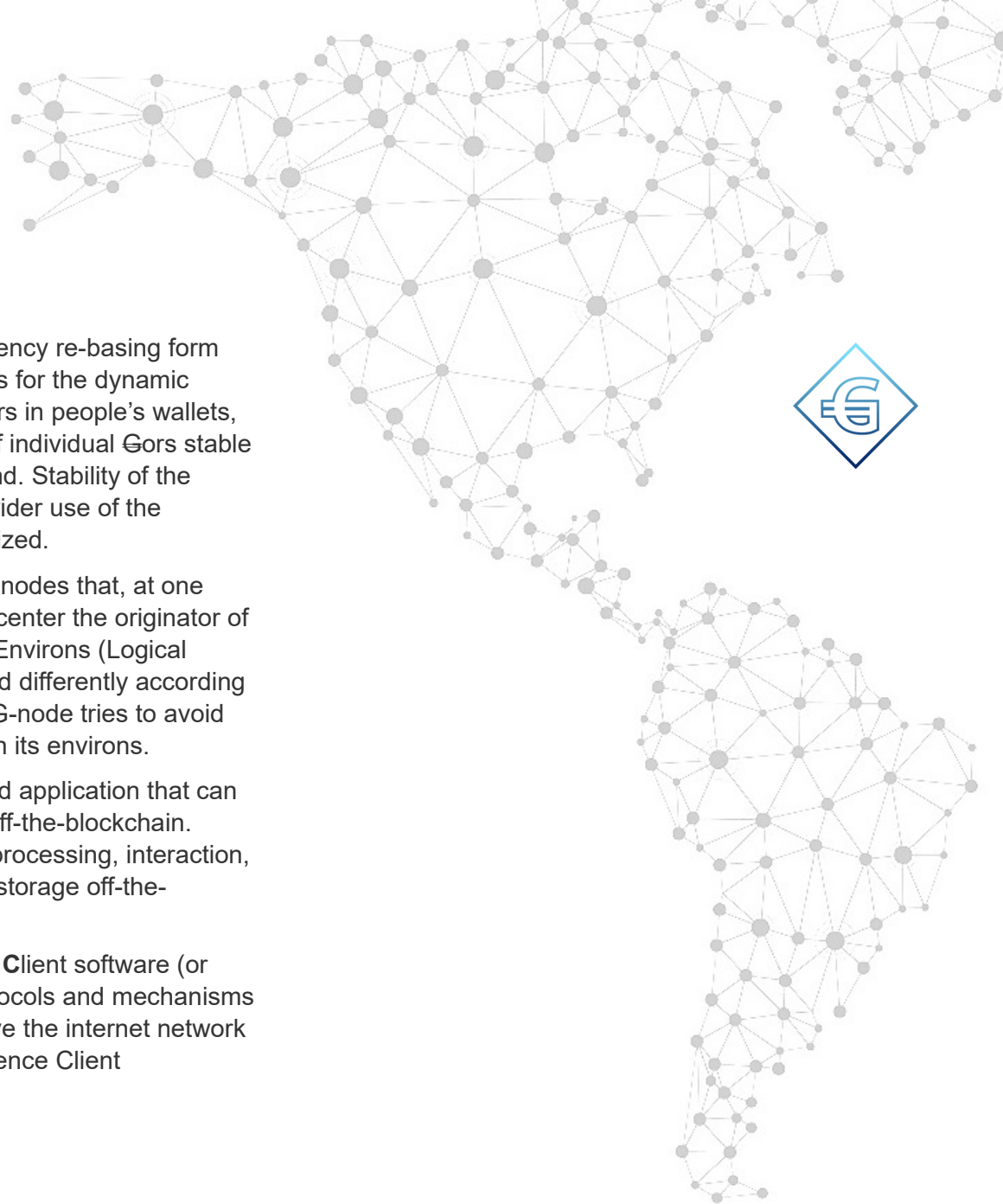
C Definitions and abbreviations

The following, are the terms and abbreviations used in this document, in alphabetical order. Normal crypto-currency terms are not necessarily included.

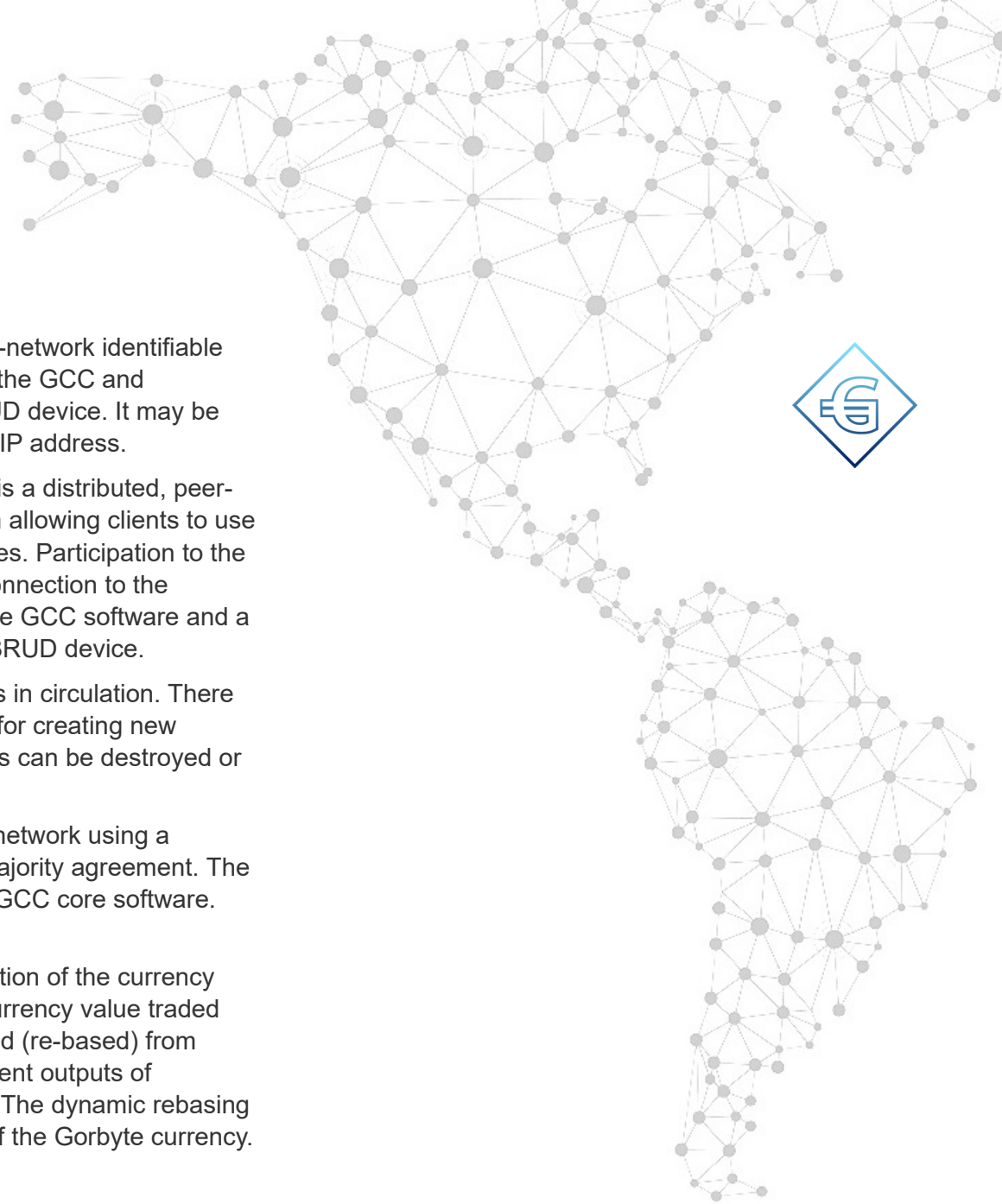
Appreciation	The acquisition of value in a currency. Currency appreciation (or deflation) is the opposite of currency devaluation (or inflation). Appreciation occurs mainly through increased currency demand. Appreciation also occurs when basegors are destroyed, for example for the purpose of sharing the proceeds from fees paid for the execution of general distributed applications (GApps).
basegors (base gors)	Not to be confused with **~~G~~ors** (the re-based Gorbyte currency that users see in their wallets – see also: Elasticity). basegors are created in the origin Gorbyte Block and are used as a unit of value in transactions and thus recorded on the blockchain.
Block	A series of transactions (Tx's) describing a financial operation, a log, or an object of a distributed application . Tx's are stored in the blockchain. Blocks are not normally shared between G-nodes. They are shared only during initialization and synchronization of nodes.

Block period	The period of time that includes the time for assembling transactions into the current block and reaching a consensus agreement about the block, before the next block can start being assembled.
Blockchain	A series of assembled blocks permanently stored in a node's database, replicated on each node of the Gorbyte network, and unchangeable (i.e. if it is changed it can be proven invalid).
BRDG device	A wearable BRUD device. It acts as a G-node, contains a blockchain replica, securely manages encryption keys, allows for unique identification of the user, includes user biometrics, acts as a hardware wallet, allows for superconnectivity and is tamper-proof.
BRUD device	A Blockchain-Registered Unique Device. A virtual unit, or USB device or an autonomous hardware device. It is used, among other purposes, to uniquely identify nodes and prevent Sybil, DoS and majority attacks. It will include biometric functionality, communication functionality, will run a full Gorbyte node and will be used as a tamper-proof hardware wallet (See "The BRUD architecture" document).
Client software	The Gorbyte software (or code) that implements a node and runs in a user device connected to the internet. See GCC. Also called the "Reference Client implementation".

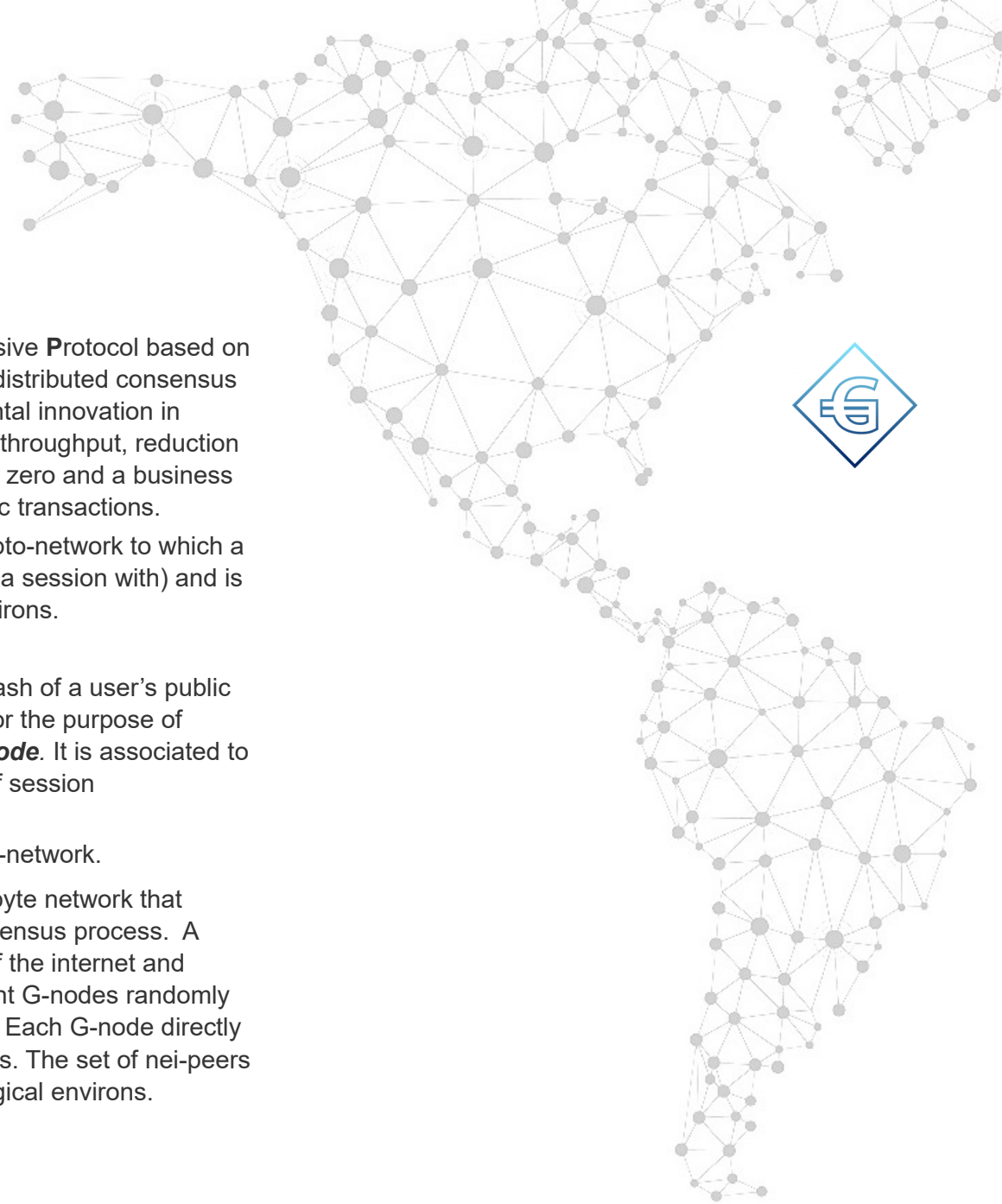
Consensus	The process or mechanism by which a majority agreement is reached regarding the exact composition of a Block to be stored in each node's Blockchain.
DApp(s)	Dapps are a limited set of distributed applications (Smart Contracts). They are objects stored, running, and producing results strictly on the blockchain.
Devaluation	The loss of value in a currency. Currency devaluation (or inflation) occurs when new currency is created. In Gorbyte no new base currency (basegors) is ever created. A currency also loses value with diminished currency demand. The opposite of "Appreciation".
DOE	The Distributed Operating Environment created by Gorbyte for running distributed applications on the blockchain. It consists of the fundamental building blocks needed to support any distributed application: Communication, addressability and identification of the players in a distributed application, guaranteed replication of data, and security.



Elasticity	The dynamic process of currency re-basing form basegors to ~~G~~ors. This allows for the dynamic change in the ***number*** of ~~G~~ors in people's wallets, thus maintaining the ***value*** of individual ~~G~~ors stable during periods of high demand. Stability of the currency value allows for a wider use of the currency, as prices are stabilized.
Environs	The logical subset of peer G-nodes that, at one moment in time, have at the center the originator of an SFI broadcast message. Environs (Logical neighborhoods) are perceived differently according to one's point of view. Each G-node tries to avoid clusters of its nei-peers within its environs.
GApp(s)	GApps are general distributed application that can use the blockchain, but run off-the-blockchain. GApps can do most of their processing, interaction, data manipulation, and data storage off-the-blockchain.
GCC	The **G**orbyte **C**rypto-network **C**lient software (or code), implementing the protocols and mechanisms of the Gorbyte network (above the internet network layer). Also called the "Reference Client implementation".

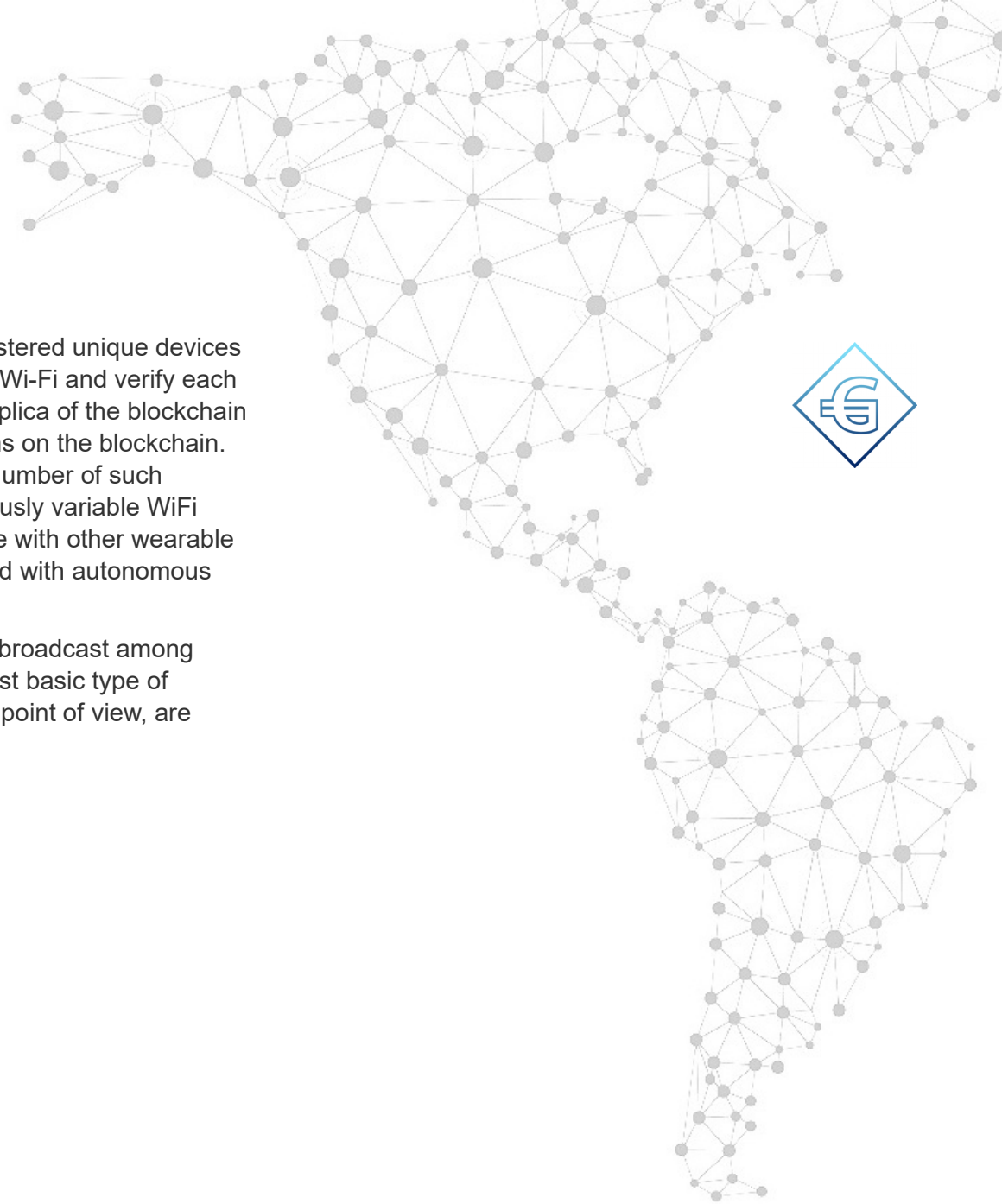


G-node (Gorbyte node)	A node of the Gorbyte crypto-network identifiable by an address generated by the GCC and associated to the user's BRUD device. It may be temporarily associated to an IP address.
Gorbyte (Gorbyte network)	The Gorbyte crypto-network is a distributed, peer-to-peer, secure, open system allowing clients to use distributed processing services. Participation to the Gorbyte project requires a connection to the internet, a working copy of the GCC software and a Virtual BRUD software or a BRUD device.
Gorbyte money supply	The total number of basegors in circulation. There is no mechanism in Gorbyte for creating new basegors. However, basegors can be destroyed or lost.
Gorbyte network	A second-generation crypto-network using a cooperative consensus by majority agreement. The set of full nodes running the GCC core software.
Gorbyte node (G-node)	See: G-node.
~~G~~ors (~~G~~)	A re-based digital representation of the currency value in a user wallet. The currency value traded between users. It is calculated (re-based) from basegors associated to unspent outputs of addresses in a user's wallet. The dynamic rebasing of ~~G~~ors allows for elasticity of the Gorbyte currency.



MARPLE	A **M**ajority **A**greement **R**ecursive **P**rotocol based on **L**ogical **E**nvirons. Gorbyte's distributed consensus protocol. The most fundamental innovation in Gorbyte, allowing scalability, throughput, reduction of network operation costs to zero and a business model providing no-cost basic transactions.
Nei-peer G-node	A G-node of the Gorbyte crypto-network to which a user node is connected (has a session with) and is part of the node's logical environs.
Node (Gorbyte node)	*See **G-node**.*
Node address	A standard address (i.e.: a hash of a user's public key) generated by the user for the purpose of temporarily identifying a ***G-node***. It is associated to a BRUD device at the time of session establishment.
Peer node	A node of the Gorbyte crypto-network.
Random network	The concept behind the Gorbyte network that allows for its distributed consensus process. A logical network built on top of the internet and consisting of physically distant G-nodes randomly connected through sessions. Each G-node directly communicates to its nei-peers. The set of nei-peers of one node is that node's logical environs.

Reference implementation	A copy of the specifications, open source code, and downloadables provided for reference.
Session	A random relation established between two physically distant G-nodes for the purpose of exchanging datagrams, transactions and MARPLE messages over underlying communication layers. A G-node and those peer nodes with which it has a session with, defines a logical environs.
Smart contract	An object on the blockchain that executes the terms of a contract. An Ethereum smart contract is a computer program addressable by its hash. It executes on the blockchain according to transactions issued to it by its owners. The output of a smart contract may be financial transactions or transactions sent as input to other smart contracts.
Summary block	Summary Blocks are those block with a block number ending with a number of zero bits, that determines the summary period's length (for example, 10 zero bits, for 1024 blocks).
Summary period	A summary period is the period of time between two summary blocks.



Superconnectivity	The ability of blockchain registered unique devices to dynamically meet through Wi-Fi and verify each other by reading their own replica of the blockchain without executing transactions on the blockchain. This allows for an unlimited number of such devices, within their continuously variable WiFi bubble, to interact in real time with other wearable devices, with IoT devices, and with autonomous robots.
Transaction (Tx)	One type of message that is broadcast among Gorbyte peer nodes. The most basic type of transactions, from the user's point of view, are financial transactions.

References and Notes

[1] “Bitcoin Covenants”, by Malte Möser, Ittay Eyal, and Emin Gün Sirer, at: http://fc16.ifca.ai/bitcoin/papers/MES16.pdf

[2] “Hayek Money: the Cryptocurrency Price Stability Solution” Ferdinando M. Ametrano, Milan Bicocca University, at: http://minco.me/SSRN-id2425270.pdf

[3] Note that the original idea of Open Distributed Processing (ODP) is not necessarily Peer-to-Peer. There are differences between a distributed application retaining centralized aspects of control and DOE-based, Peer-to-Peer distributed applications, with truly distributed control.