DAOs and BORGs: Blending the Best Trust-Minimization Techniques of the Onchain and Offchain Worlds

GABRIEL SHAPIRO

AKA @LEX_NODE

CRYPTOLAWYER

METALEX

Introduction

Exploring how DAOs and BORGs can be combined into onchain/offchain hybrid check/balance mechanisms to address common DAO trust issues & legal risks

I will go super fast through the 'foundational' material to be able to focus more on specific trust-minimization techniques—don't worry if you don't grok everything!

What is a DAO, anyway?

"The DAO that can be defined is not the true **DAO.**"
-Dao Tzu

What is a DAO, anyway? – answer #1

Best you can do if you want to honor the "DAO" label of everything that is called a "DAO":

- A DAO is any entity or organization that uses smart contracts for any of its functions (voting, holding money, etc.)
- Problem:
  - Many things called "DAOs" under this construct are either not "decentralized" or not "autonomous"
    - Ex: "Metacartel Venture DAO"
      - =compliant "Venture DAO" that is an LLC complying with Investment Company Act exemptions & holds many offchain securities:
      - capped at < 100 investors (not decentralized)
      - censorable/stoppable by the state & third parties (not autonomous)

What is a DAO, anyway? – answer #2

My quick super opinionated answer for what are “real DAOs”:

- DAOs must be both:
  - decentralized
    - any residual human discretion (i.e., intrinsic modalities of power) are systematically dispersed over a large, agile, and potentially anonymous group of incentive-aligned persons, preferably with permissionless access of third parties to acquiring that power on fair terms
  - autonomous
    - self-governing, trust-minimized and resistant to extrinsic exercises of power.

What is a DAO, anyway? – answer #2 (cont'd)

Three rules of DAObotics (adapted from **Stan** Larimer)

- A DAO can be analogized to a corporation having tokens as its shares and code as its bylaws (he called Bitcoin itself a "DAC" which allows miners (workers) to hire themselves and be in BTC as system equity shares)
- Law #1:
  - DAOs must "run under the control of an incorruptible set of rules that are implemented as publicly auditable open source software…":
- Law #2:
  - A DAO must not be able to change its rules without consent of its stakeholders and such consent must not violate law #1
- Law #3:
  - A DAO must protect its own existence, as long as such protection does not conflict with law #1 or law #2.

What is a BORG?

- a cy**B**ernetic **ORG**anization
- = a traditional legal entity incorporated with a legal requirement to use decentralized or autonomous technologies (such as smart contracts or AI) to augment all or a portion of the entity's governance and activities
- The BORG concept represents a blend of legal and technological governance, similar to cyborgs merging human and robotic functions.
- BORGs vary in form, from tech-augmented corporations with tokenized shares to DAO-adjacent organizations offering emergency multisig support or grants-giving functions.

BORG Types

DAO-adjacent BORGs

- trust-mitigated, accountable, DAO-adjacent entities, such as a Foundation that wraps an emergency multisig for an DeFi protocol, but gives the DAO on-chain control over the emergency multisig's powers (eg, can veto appointment/removal of signers or revoke the multisig's powers entirely) and certain legal rights over the multisig signers if they abuse their power.
- Example
  - Curve Emergency BORG

BORG Types

bizBORGs

- tech-augmented companies, such as a corporation with tokenized, programmable shares (eg, tokenized preferred stock that embeds a complex set of liquidation and dividend logics)
- Example:
  - Metacartel Ventures (=a Delaware LLC structured as a compliant exempt investment company)

Why is a BORG not a DAO?

- Lacks autonomy
  - Violates DAObotics Rule#1:
    - At least some of its rules are offchain/legal rather than onchain/code-embedded
  - Violates DAObotics Rule #2:
    - Rules can change without consent of stakeholders (example, Delaware can revise its corporate code without approval from the stockholders of a Delaware corporation, and this can change the stockholders' rights)
  - Violates DAObotics Rule #3:
    - Often has its own extinction as a possibility or even goal (i.e., get acquired and be the 'disappearing corporation' in a profitable merger that serves as a liquidation event for stockholders)
- Lacks decentralization:
  - May be owned/controlled by a small and/or 'unfairly' permissioned set ("close corporations", Foundations managed by a modestly sized board of directors, etc.)

Side-note: Why BORGs and not subDAOs?

- subDAOs are usually not ‘DAOs’ but rather small multisigs, hence the term “DAO” is inappropriate (not decentralized, also often not autonomous)
- DAO is in stronger control of the entity and hence DAO is likely to be a liability target (or stronger liability target) if subDAO does something ‘wrong’.
- Securities laws—don’t want DAO tokens to be similar to shares in a business entity, which they can be if they have too much control over a business entity styled as a ‘subDAO’

Trust Problems re: DAOs, DAO-adjacent entities & their counterparties

Remember: verify, don't trust is the ethos of crypto

Trust Problems – member management

- Member management
  - DAO may approve initial membership composition of a DAO-adjacent entity (e.g., a protocol security multisig or BoD of a foundation) but has to trust-not-verify that:
    - the people who end up with multisig power actually are the ones approved by the DAO
    - that multisig members manage their own membership appropriately, either seeking approval from the DAO for new members or removing members who the DAO wants removed, or quality-controlling membership and members' conduct without input from the DAO but still aligned with the DAO's interests
    - This issue is particularly bad if the initial trust assumption of the DAO was that 'reputation slashing' is a sufficient check on the entity—if the members shift from high-reputation to low-reputation over time, the DAO loses its 'reputation slashing' deterrent/punishment!
  - A member who wishes to 'resign' (liability concerns, life concerns, wrench attack concerns, whatever) must trust the other members of the multisig to 'vote them out'

Trust Problems – asset management

- DAO may directly or indirectly contribute significant tokens or value to a DAO-adjacent entity (e.g. a grants multisig or a foundation) but has to trust-not-verify that the tokens/value are used in the DAO's/community's interests
- Asset management transparency
  - The DAO may think that because funds are initially in a multisig that there is transparency, but the DAO must trust-not-verify this is so, because in theory a DAO-funded multisig that initially is transparent to the DAO onchain can
    - move tokens into a CEX or custodian and run them opaquely
      - example—DeFi Education Fund moved most of its UNI out of the multisig shortly after UNI was funded, despite repeated transparency promises, then didn't even post significant voluntary reports for a long time (no external auditor)
    - cash out all its tokens into fiat and run things from an opaque bank account.
    - use a mixer and switch funds to a new private multisig
- Asset management accountability
  - The DAO must trust-not-verify that assets are used as contemplated by the original DAO proposal for the funds grants
  - Proper usage can be difficult to verify and depend on offchain facts
  - If improper usage occurs, the DAO may have no or few practical remedies for holding the entity's managers accountable (other than 'reputation slashing,' which may be weak as subjective faults have more deniability / lower attributability & 'cancel culture' does not always work)

Trust Problems – permissions management

- A DAO-adjacent entity may receive specific privileges or permissions (e.g., ability to “freeze the protocol”) that are only supposed to be used for specific reasons (e.g., to prevent or mitigate a hack), but the DAO must trust that these powers are in fact so used
- Example
  - Kujira multisig members changed liquidation thresholds and froze liquidations and eventually the entire protocol to save the dev team’s own leveraged positions in the protocol

Trust Problems – Bad Mitigations

- One way of addressing these issues is to always make everything that the DAO-adjacent entity has revocable and controllable by the DAO, but then the trust problem is flipped too far the other way—the DAO-adjacent entity workers are fully at the mercy of the DAO, and DAOs don't make great counterparties:
  - Lapse into subDAO model with attendant legal issues
  - There is likely to be an 'adverse selection effect' among potential members/managers of DAO-adjacent entities, as people do not want to work for whimsical counterparties that can rug them arbitrarily
    - No employment law protections (severance, unemployment benefits, etc.)
    - No contractual protections (if you build X we will pay you Y and you can sue us if we don't)
  - Example:
    - Juno DAO got cold feet and rugged its entire complex subDAO system after contributors' spent months of their lives setting it up, before it even really had a chance to get going

Trust Issues Faced by Extrinsic Counterparties

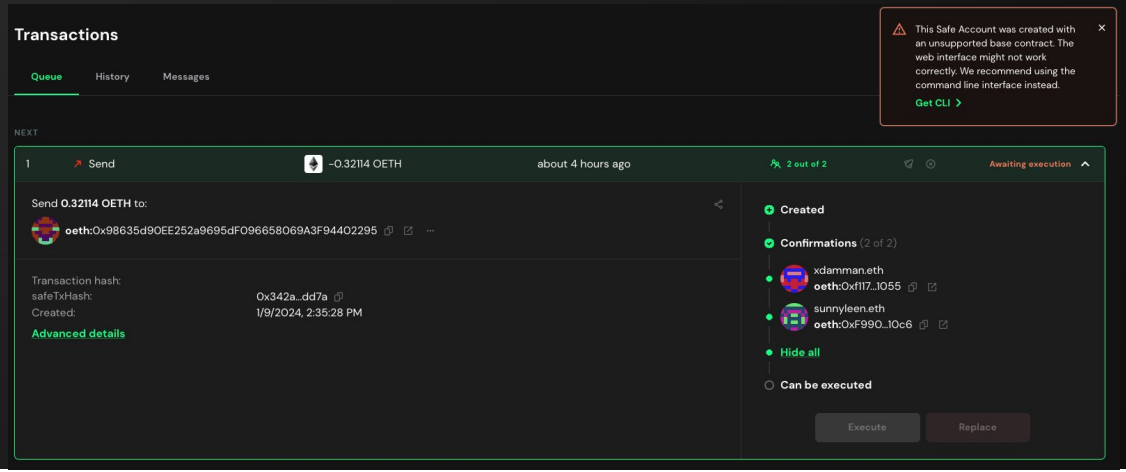
- DAOs, subDAOs and naively structured DAO-adjacent entities make terrible counterparties for third parties
  - I know of one tokenomics consulting company that has a ‘no DAO policy’
  - Many law firms also will not/cannot represent DAOs per se
  - Why?
    - DAO, subDAO or DAO-adjacent entities often lack legal structure
    - Can in practice rug the counterparty at any time for any reason or no reason due to use of simple / naïve onchain structures like naked SAFEs or naked DAO custody
    - Counterparty may fully perform their contract but never receive the reward and have no practical remedy (cannot sue or even if can sue, cannot actually get money out) . . . there is no remedy, only potential ‘punishment’ in the form of reputation slashing (which itself has weak subjective attributability and uncertain impact)
  - This dynamic is likely to lead to adverse selection effects, not to mention unfair outcomes! (the latter tend to result in the former)

Addressing Trust Problems w/ BORGs

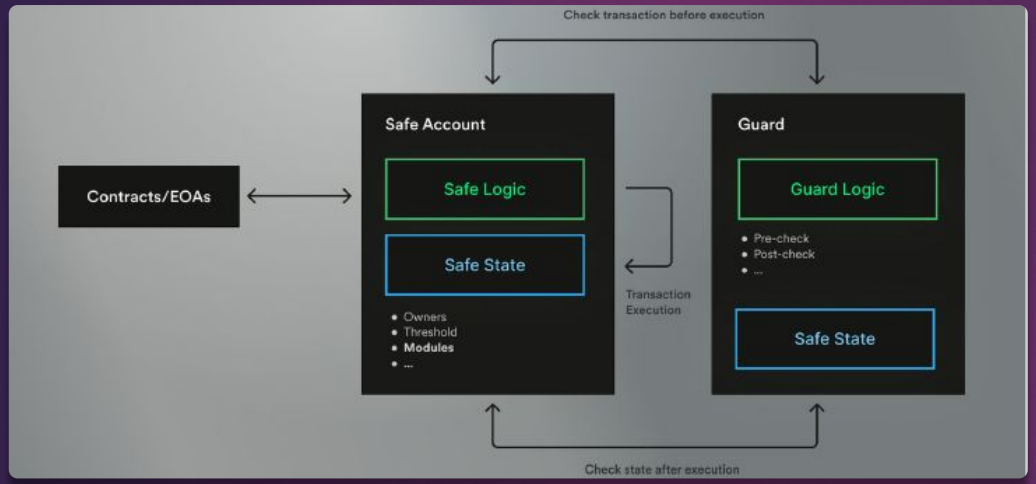
- Onchain Techniques
- Offchain Techniques

Addressing Trust Problems – Onchain Techniques

- SAFE is the multisig solution of choice on EVM and has inbuilt hooks for ‘guards’ and ‘modules’
- We can modulate the functionality of a SAFE to constrain the discretion of its members and impose a ‘can’t be evil’ philosophy



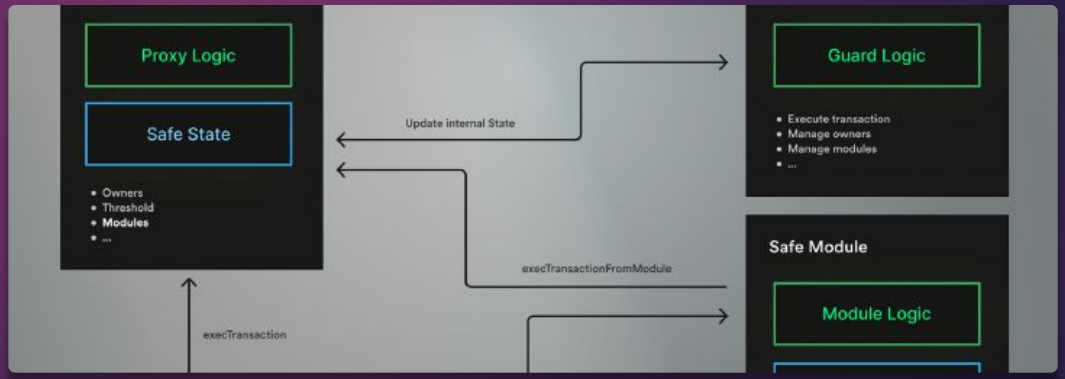
Addressing Trust Problems – Onchain Techniques - Guards



GUARDS CONSTRAIN SAFES (E.G. CAN MAKE SURE ‘WRONG’ TRANSACTIONS DON’T EXECUTE)

Addressing Trust Problems – Onchain Techniques - Modules

- Modules expand SAFEs (e.g. can allow a DAO to execute a SAFE function outside of normal approval flow)



Addressing Trust Problems – Onchain Techniques - Implants

Three ‘Modes’ for BORG SAFEs:

- Whitelist
  - all transactions are ***prohibited***–except those explicitly permitted (i.e., whitelisted).
  - Only whitelisted contracts can be interacted with, and each whitelisted contract's methods can have specific parameter constraints (including types (uint, int, address, string, bytes, bool), value ranges, and exact matches).
  - Whistlisting can be done through external governance
    - A DAO
    - Another SAFE
    - etc.
  - This BORG mode is most suitable for BORGs handling large amounts of money–for example,
    - a finBORG that is managing ‘protocol-owned’ or ‘protocol-beneficial’ value by moving liquidity among various whitelisted liquidity pools and DeFi protocols
    - a Grants BORG that is subject to strict rate-limits or DAO vetoes or DAO co-approvals.

Addressing Trust Problems – Onchain Techniques - Implants

Three ‘Modes’ for BORG SAFEs:

- Blacklist
  - In blacklist mode, all transactions are ***permitted***–except those explicitly pre-prohibited (i.e.,blacklisted)
  - Blacklisted contracts cannot be interacted with, or methods of these contracts are blocked unless they pass specified parameter constraints (including types (uint, int, address, string, bytes, bool), value ranges, and exact matches)
  - Blacklist mode is suitable for DAO-adjacent BORGs that are somewhat more trusted (or somewhat less risky, and therefore less trust-requiring) and therefore may have broad discretion and flexibility, but with respect to which the DAO wishes to constrain a short list of particularly risky, dangerous or prohibited transactions. For example, a group of friends managing a memecoin and is trusted to use presale proceeds well, but which requires a snapshot approval of the memecoin holders in order to increase the memecoin supply through a minting function on the token smart contract. In this case, the mint() function on that particular token contract would be blacklisted, and implants (explained below) would be used to only allow a mint() call if also approved by the required snapshot vote.
- Free
  - Unrestricted mode is suitable for highly ‘trusted’ BORGs where the multisig signers are intended to have full discretion over the BORG’s activity, but where there is a desire to use some modules (see below)

Addressing Asset Management Trust Problems – Onchain Techniques - Implants

- OptimisticGrantImplant.sol
  - Enables a GrantsBORG in ‘whitelist mode’ to give out grants “optimistically” using funds granted by the adjacent DAO, subject to programmatic rate limitations and caps.
  - The GrantsBORG may exceed the rate limitations and/or caps
    - with DAO-co-approval or
    - subject to a timelock + absence of DAO veto within timelock period

Addressing Asset Management Trust Problems – Onchain Techniques - Implants

- daoVETOGrantImplant.sol
  - Enables the aforementioned timelock + DAO veto pattern for exceptions to the GrantsBORG's rate limits and caps.
  - Optional: anti-DoS measures so that the BORG cannot overwhelm the DAO's veto capacity via 'spam' proposals (cooldown period b/w SAFE transaction proposals that are subject to the veto).
- daoVoteGrantImplant.sol
  - Enables the aforementioned DAO co-approval pattern for exceptions to the GrantsBORG's rate limits and caps.

Addressing member management trust issues– Onchain Techniques

- ejectImplant.sol
  - Enables
    - a GrantsBORG multisig signer to resign of their own accord or
    - the DAO to remove a GrantsBORG multisig signer
  - Imagine hybrid arrangements
    - DAO and existing multisig members need to co-approve new multisig members
    - But DAO alone can ‘remove’ a member (e.g., because of egregious behavior and other multisig signers are too scared of lawsuits to remove the bad signer)
- failSafe.sol
  - Enables funds to revert to the DAO (or another address) on specified events—for example, if the number of BORG signers falls below the minimum threshold needed for approval of actions on the SAFE (this can happen when you enable resignation or removal via the other functions!!!) .
  - The destination address (typically the DAO treasury) is set at deployment and immutable. Like with other Implants, this can be combined with the
  - A ConditionManager to allow for clawbacks triggered by DAO approval alone, or DAO approval + number of other conditions or approvals.

Addressing Trust Issues – Offchain Techniques

- Have a Legal Wrapper for your DAO-adjacent BORG,
  - Why?
    - Limited liability for participants
    - Continuity for DAO (example: entity owns domain names, IP, and can be preserved continuously under one entity name even though personnel identities keep changing over time)
    - Better counterparties for contributors and partners (they have an entity with assets to sue if they are treated unfairly, and this entity is separate from DAO so does not mean suing this entity potentially destroys the DAO from a legal/regulatory perspective)
  - How?
    - Entity legally owns the SAFE and its assets
    - People who are on the SAFE, work for/manage the entity
      -

Addressing Trust Issues – Offchain Techniques

Choose Type of Legal Wrapper Wisely

- ***Note:*** *these considerations for DAO-adjacent BORGs; for bizBORGs corporations, LLCs, etc. are most suitable*
- corporations (Inc.) and companies (LLC)
  - designed to have equity owners to whom the entity's managers are ultimately accountable (usually to generate a profit on investment)
  - if your DAO-adjacent BORG is a corp or company, who will own it and why do you want the interests of those owners to be superior to the interests of the DAO/community? is it even possible for the entity's managers to ***consider*** the interests of the community/DAO in decisionmaking or are they legally ***required*** to only benefit the owners?
  - In most states LLCs can waive most duties to members through custom drafting in the operating agreement; some states have "DAO LLCs" that do this by default—however, members may retain ultimate authority on some topics (like M&A transactions) with no default limitation on selfish voting
  - Can potentially address issue by making DAO tokenholders into stockholders/LLC members but this may create major securities law/tax law and other compliance concerns, especially in the EU where securities status tends to hinge on the distinction between contract-rights-bearing tokens and non-contract-rights-bearing tokens
  - Usually taxed
  - Example
    - dYdX labs company switched from being an ordinary DE corporation to a 'public benefit' DE corporation in order to be ***permitted*** to take into DAO/community interests; however, this is still not ***required*** nor is there any remedy for the DAO/community if their interests are ignored or subordinated to the interests of stockholders

Addressing Trust Issues – Offchain Techniques

- foundations
  - Cayman is best because there is no ***default or non-waivable beneficiary/member/founder structure***
    - No members/owners
    - No beneficiaries
    - Almost infinitely flexible management structure (can create bespoke roles and define them with whatever powers and interrelationships you want)
  - Other jurisdictions (Panama, Switzerland) fit less cleanly because they have more non-waivable defaults (like having a beneficiary in the case of Panama)
  - Often tax-free, at least on paper (can mess it up with personnel from wrong jurisdictions)
- Misc. – vereins, partnerships etc.
  - Can work but usually not ideal for various reasons

Addressing Trust Issues – Offchain Techniques

Define Your Entity's Purposes Wisely

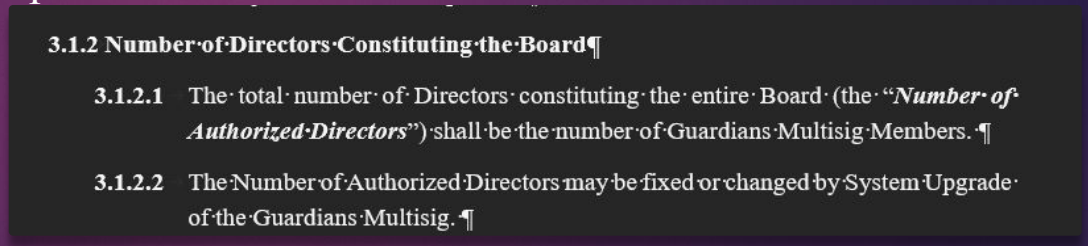
- Bad structure:
  - 'follow the will of Lido DAO';
  - 'benefit LDO token holders'
- Good structure
  - 'give Firestarter style grants over three years, which primarily benefit the Lido community o one or more sub-constituencies thereof'
  - Lido community would be defined as broader than just DAO (includes LDO, stETH, validators, users, other protocols etc.)
  - Can establish weightings among competing interests (e.g., put stETH holders #1, then LD holders #2, or whatever…different priorities make sense for different BORGs!!! A securi BORG should prioritize interests of users wi TVL in the protocol)

- **Permitted Purposes**
  - The permitted purposes of the BORG (the "***Purposes***") are to:
    - on a purely volunteer *ad hoc*, episodic basis and without obligation or liability for continuity, promptness, or quality of efforts, utilize the Emergency Multisig's permissions over the Community Autonomous Systems, to prevent, stop, limit, defend against, or mitigate the adverse effects of, any Security Threats (such uses of the Emergency Multisig, the "***Authorized Uses***");
    - operate and secure the Emergency Multisig;
    - adhere to and enforce the provisions of these Bylaws;
    - hold and use any Blockchain Tokens lawfully owned by the BORG in support of the upkeep and maintenance of the BORG (including its continued corporate good standing) and the pursuit of the Authorized Uses; and
    - to do all such other things as are or may be incidental or conducive to any or all of the above-referenced purposes.
  - "***Security Threats***" means: (a) any actual or reasonably expected, threatened, imminent, pending or ongoing frauds, thefts, misappropriations, extortions, abuses, hacks, attacks, exploits, intrusions, abuses, malicious manipulations, griefings, denials-of-service, or adversarial freezings or impairments, or other similar misconduct against or dysfunction involving any, of the Community Autonomous Systems (or any software or system on which any of the Community Autonomous Systems depends in whole or in part), in each case that could reasonably be expected to result in disruption or impairment of, harm to, misappropriation of other adverse effects on or damage to the Community Autonomous Systems, any Blockchain Tokens created or constituted or held thereby or thereon or under the control or custody thereof, or any members of the Community (collectively, "***Security Incidents***"); and (b) any bugs, defects, or errors in the Community Autonomous Systems (or any software or system on which any of the Community Autonomous Systems depends in whole or in part) that could reasonably be expected to lead to a Security Incident

Addressing Trust Issues – Offchain Techniques

Make a SAFE your canonical Board of Directors hub

- Every SAFE member is a Director, and every Director is a SAFE member—legally enforced (tight mirroring)
- This means direct, verifiable management structure onchain, with potential direct influence from DAO through the implants mentioned above



3.1.2 Number of Directors Constituting the Board

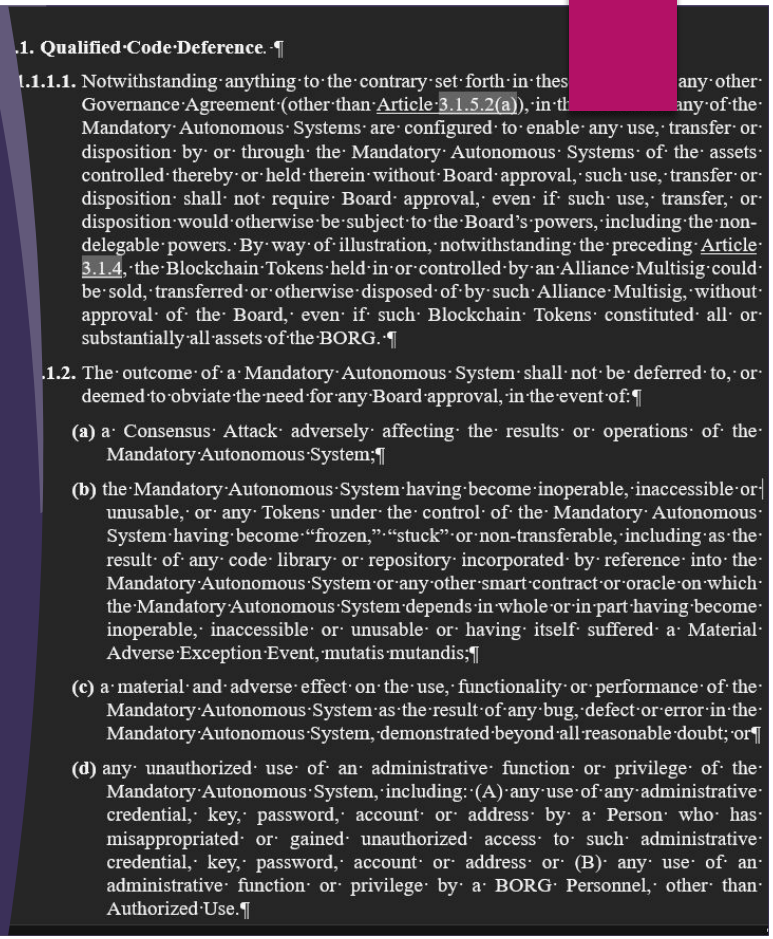
3.1.2.1 The total number of Directors constituting the entire Board (the "***Number of Authorized Directors***") shall be the number of Guardians Multisig Members.

3.1.2.2 The Number of Authorized Directors may be fixed or changed by System Upgrade of the Guardians Multisig.

Addressing Trust Issues – Offchain Techniques

Qualified Code Deference Provision

- This is basically a rule that enshrines the onchain trust-minimization techniques discussed above into the legal rules, by reference
- So basically if the SAFE used as the BORG's funds management tool imposes a rate-limitation onchain, that is automatically mandatory for the legal entity itself
  - Makes clear that some of the legal rules are embedded in code
  - Makes code end-runs (in case there are any) illegal – code is never perfect so this is a nice hedge
- Why "qualified"?
  - In edge cases the code may be configured to violate a law or may have bugs. But try to keep the 'qualification' narrow to maximize benefits of code and minimize likelihood of disputes.



.1. Qualified Code Deference. ¶

.1.1.1. Notwithstanding anything to the contrary set forth in thes any other Governance Agreement (other than Article 3.1.5.2(a)), in th any of the Mandatory Autonomous Systems are configured to enable any use, transfer or disposition by or through the Mandatory Autonomous Systems of the assets controlled thereby or held therein without Board approval, such use, transfer or disposition shall not require Board approval, even if such use, transfer, or disposition would otherwise be subject to the Board's powers, including the non-delegable powers. By way of illustration, notwithstanding the preceding Article 3.1.4, the Blockchain Tokens held in or controlled by an Alliance Multisig could be sold, transferred or otherwise disposed of by such Alliance Multisig, without approval of the Board, even if such Blockchain Tokens constituted all or substantially all assets of the BORG. ¶

.1.2. The outcome of a Mandatory Autonomous System shall not be deferred to, or deemed to obviate the need for any Board approval, in the event of: ¶

(a) a Consensus Attack adversely affecting the results or operations of the Mandatory Autonomous System; ¶

(b) the Mandatory Autonomous System having become inoperable, inaccessible or unusable, or any Tokens under the control of the Mandatory Autonomous System having become "frozen," "stuck" or non-transferable, including as the result of any code library or repository incorporated by reference into the Mandatory Autonomous System or any other smart contract or oracle on which the Mandatory Autonomous System depends in whole or in part having become inoperable, inaccessible or unusable or having itself suffered a Material Adverse Exception Event, mutatis mutandis; ¶

(c) a material and adverse effect on the use, functionality or performance of the Mandatory Autonomous System as the result of any bug, defect or error in the Mandatory Autonomous System, demonstrated beyond all reasonable doubt; or ¶

(d) any unauthorized use of an administrative function or privilege of the Mandatory Autonomous System, including: (A) any use of any administrative credential, key, password, account or address by a Person who has misappropriated or gained unauthorized access to such administrative credential, key, password, account or address or (B) any use of an administrative function or privilege by a BORG Personnel, other than Authorized Use. ¶

Addressing Trust Issues – Offchain Techniques

Emergency Supervisor

- With a foundation, you can create a special role called the ‘emergency supervisor’
- This would be an entity or individual appointed by the DAO in a case where the DAO suspected the BORG had broken the BORG’s own rules
- Emergency Supervisor has combined ***statutory*** and ***contractual*** mandate to investigate wrongdoing, sue people in the name of the BORG, remove and replace wrongdoers, etc. (can expand or contract these powers as desired, making Emergency Supervisor more or less powerful)
- Critical, the point is not for the Emergency Supervisor to ‘manage the entity generally on behalf of the DAO’ but simply to make sure that the BORG is following the BORG’s own rules (which the DAO likely previously approved at inception)

2.6. *Emergency Supervisors*

2.6.1. Purposes and Powers.

2.6.1.1. The Emergency Supervisor shall have the mandate to enforce th ORG and act in the name of and represent the BORG and to bring claims i e BORG, in each case, solely to the extent necessary or desirable to handle the applicable Adverse Event. The powers of the Emergency Supervisor shall include the power to, to the extent necessary or desirable handle the applicable Adverse Event:

(a) appoint an interim Director in accordance with Article 2.2.3.2(b);

(b) remove any one or more Director(s), Multisig Members or other BORG Personnel who have committed or knowingly assisted in the commission or furtherance of an Adverse Event; or

(c) initiate and pursue legal proceedings by, on behalf of or in the name of the BORG against one or more Director(s), Multisig Member(s) or other BORG Personnel who have committed or knowingly assisted in the commission or furtherance of an Adverse Event,

in each case, subject to the requirement that the Emergency Supervisor, in acting for, on behalf of, at the direction of, or using the resources of the BORG and in accordance with the contractual provisions of these Bylaws, shall set aside their direct and indirect personal interests, and shall solely act in furtherance of the Purposes in accordance with the Principles.

2.6.1.2. The Emergency Supervisor shall observe, implement, carry out, action, and execute any and all Community Module Approvals that are lawful, reasonable, and made in good scope, and are within the scope of the Emergency Supervisor’s authority, with commercially reasonable best efforts and in a commercially reasonable and timely manner and in a manner not inconsistent in any material respect with the Community Module Approval(s) appointing and instructing the Emergency Supervisor; provided, however, that the Emergency Supervisor shall not be required to expend any of its own funds or incur any liabilities in performing its duties.

2.6.2. Appointment.

If there has been an Adverse Event, then, solely to the extent necessary or desirable to investigate, resolve, hold persons liable for, or otherwise handle such Adverse Event or its consequences, the Community Module may appoint by Community Module Approval one or more additional Supervisors specifically mandated for such purposes (an “***Emergency Supervisor***”). A person appointed by the Community Module as Emergency Supervisor must affirmatively accept the role of Emergency Supervisor by written notice to the Community Module (e.g. by publishing such notice on one or more public URLs or social media channels known to and accessible by the Community) and the BORG within 30 days after the date of the relevant Community Module Approval, and such appointment shall be deemed automatically effective on the date of the last such notice delivered. Failure to affirmatively accept such role as set forth above shall automatically be deemed a rejection of such role on the 30th day after the relevant Community Module Approval. Once the Adverse Event has been handled in accordance with all applicable Community Module

Addressing Trust Issues – Offchain Techniques

Amendment Rules

-Require DAO co-approval for changing any legal rules of the BORG in a way that could adversely affect the DAO or broader community. This prevents ‘legal rug-pulling’ by amending-out provisions (like the emergency supervisor authorization) that are intended to protect the DAO/community

5. CERTAIN COMMUNITY MODULE APPROVALS

In addition to any other matters calling for Community Module Approval as set forth in these Bylaws, the following matters shall require a prior Community Module Approval, in addition to approval by the Board (and any other approval that may be required under these Bylaws, the Constitution or any other Governance Agreement):

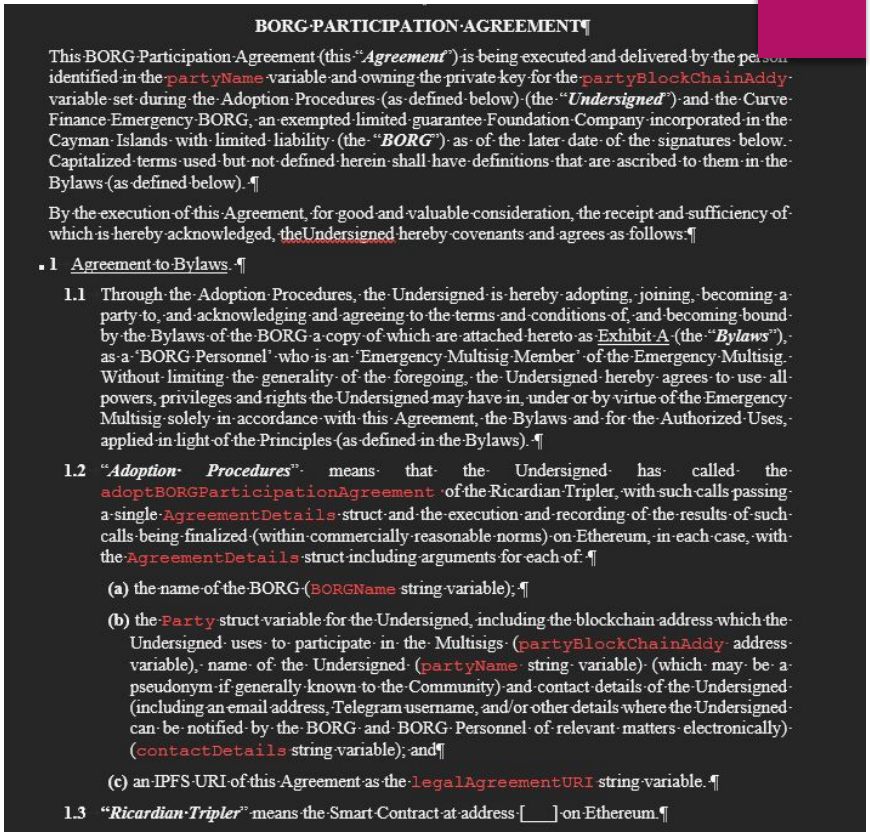
5.1. any Liquidation Event;

5.2. any amendment, addition, deletion, modification, waiver, or change to any one or more of the Purposes, Authorized Uses, or any other provision of these Bylaws referring to the Community (or any sub-constituencies thereof), the Community Module, any Community Module Approval or any Community Module Veto, whether to be approved via amendment to these Bylaws, any other Governance Agreements, or otherwise, where such amendment, addition, deletion, modification, waiver, or change, whether individually or in the aggregate with other amendments, additions, deletions, modifications, waivers, or changes or existing provisions could reasonably be expected to negatively modify, limit, eliminate, waive, or otherwise adversely affect any power, right, obligation, liability, perquisite, or interest of the Community (or any one or more sub-constituencies of the Community); and

Addressing Trust Issues – Offchain Techniques

Ricardian Triplers

- Enable onchain agreements so that the DAO is not reliant on the BORG's private database to know what legal agreements the BORG owns (this helps ensure continuity over time even as personnel changes may occur)



BORG PARTICIPATION AGREEMENT

This BORG Participation Agreement (this "***Agreement***") is being executed and delivered by the person identified in the `partyName` variable and owning the private key for the `partyBlockChainAddy` variable set during the Adoption Procedures (as defined below) (the "***Undersigned***") and the Curve Finance Emergency BORG, an exempted limited guarantee Foundation Company incorporated in the Cayman Islands with limited liability (the "***BORG***") as of the later date of the signatures below. Capitalized terms used but not defined herein shall have definitions that are ascribed to them in the Bylaws (as defined below).

By the execution of this Agreement, for good and valuable consideration, the receipt and sufficiency of which is hereby acknowledged, theUndersigned hereby covenants and agrees as follows:

1 Agreement to Bylaws.

1.1 Through the Adoption Procedures, the Undersigned is hereby adopting, joining, becoming a party to, and acknowledging and agreeing to the terms and conditions of, and becoming bound by the Bylaws of the BORG a copy of which are attached hereto as Exhibit A (the "***Bylaws***"), as a 'BORG Personnel' who is an 'Emergency Multisig Member' of the Emergency Multisig. Without limiting the generality of the foregoing, the Undersigned hereby agrees to use all powers, privileges and rights the Undersigned may have in, under or by virtue of the Emergency Multisig solely in accordance with this Agreement, the Bylaws and for the Authorized Uses, applied in light of the Principles (as defined in the Bylaws).

1.2 "***Adoption Procedures***" means that the Undersigned has called the `adoptBORGParticipationAgreement` of the Ricardian Tripler, with such calls passing a single `AgreementDetails` struct and the execution and recording of the results of such calls being finalized (within commercially reasonable norms) on Ethereum, in each case, with the `AgreementDetails` struct including arguments for each of:

(a) the name of the BORG (`BORGName` string variable);

(b) the `Party` struct variable for the Undersigned, including the blockchain address which the Undersigned uses to participate in the Multisigs (`partyBlockChainAddy` address variable), name of the Undersigned (`partyName` string variable) (which may be a pseudonym if generally known to the Community) and contact details of the Undersigned (including an email address, Telegram username, and/or other details where the Undersigned can be notified by the BORG and BORG Personnel of relevant matters electronically) (`contactDetails` string variable); and

(c) an IPFS URI of this Agreement as the `legalAgreementURI` string variable.

1.3 "***Ricardian Tripler***" means the Smart Contract at address [___] on Ethereum.