A governance layer for the Internet

METAGOV
A governance layer for the Internet

A software framework and data model for online governance, combined with the usability and modularity of an app store
How it works

Metagov is a lightweight protocol, packaged as an extensible plugin for online platforms, that helps users govern social groups and solve collective action problems. It’s built on four principles:

1. **Modularity**: Platform operators and community members should have the ability to construct governance systems by creating, importing, and arranging component parts together as a coherent whole.

2. **Generality**: The governance layer should be possible to implement in a variety of technical contexts, so that all platforms share a consistent underlying framework for governance.

3. **Cross-platform portability**: Governance tools developed for one platform should be portable to another platform.

4. **Cross-platform interoperability**: Governance systems should have the ability to interact across platforms, sharing data and influencing processes.
**Overview**

Cross-platform toolset for governance design

World-class R&D team

Committed industry partners

Built for social networks, blockchains, online games, and messaging platforms

---

**Toolset**

| Agreement engine for contracts and orgs | Flexible smart and "dumb" contracts | Modular framework allows fast deployment | Visual programming language for module design and simulation | Cross-platform governance protocol |

---

**Research Strategy**

| Ontology of governance structures | Modular Politics journal article | University and foundation collaborations | Quick iteration through prototypes | Everything is open-source |
Imagine an Internet where **democracy** is at least as available as **autocracy**.
Governance complexity is a barrier to sustained growth.

Governance is necessary to growth. Ungoverned spaces like 4chan or Gab quickly become toxic. But governance at scale is complex and expensive.

To succeed and grow, online communities must use a diverse array of tools (laws, norms, markets, and code) to channel user behavior.

Governance is an unaddressed need

There is huge interest in decentralized governance, with experiments across Facebook, Twitter, and the blockchain. A recent example: the explosion of mutual aid groups in response to COVID-19.

However, the current solutions are ad hoc, limited, and ill-equipped for governance innovation—often featuring rigid hierarchies, static toolsets, and typologies suited more to nations than the needs of online communities.
A simple idea: empower users

Platform operators are not the only ones working on governance; users and independent developers are too.

Metagov integrates these three processes, taking the load off platforms and increasing the efficiency of the entire system.
Simple building blocks

Users form *agreements*, which are interactive virtual machines that support both textual contracts and simple computations.

*Modules* provide rich functionality for governance, including enforcement. For example: voting, tokens, reputation, and chat.

Metagov
The Metagov Stack

Governance Editor
Easy-to-use interface helps users create new agreements and modules, including from templates.

Agreement Engine
Software that initializes agreements as virtual machines and handles module installation and collision.

The Metagov Protocol
A robust computational syntax for governance that works across different online platforms.
Ecosystem

Engagement for platforms

Platforms install an Agreement Engine plugin as part of their platform, or interact with Metagov’s servers through an API. Implementation may vary in scope.

In doing so, they increase retention and engagement by allowing users to customize their experience within certain constraints. They also drastically reduce customer service costs when users are empowered to deal with their own problems.

Freedom for users

Users use the Metagov interface to manage and customize governance decisions.

Allowing users to customize aspects of the platforms they use turns them from users to co-designers.

Support for developers

Developers and scientists use the SDK and online development environment to design, test, and simulate complex social mechanisms before deploying to the Metagov network.

Built-in platform integrations allow developers to build fast and scale quickly to multiple platforms.
A portable, cross-platform tool

Our key innovation is the development of a truly cross-platform governance protocol, one that works across social networks, blockchains, games, and instant messaging.
Team
Team

Lawrence Lessig
Principal Investigator

Roy L. Furman Professor of Law and Leadership at Harvard Law School
Founder, Creative Commons
Team

Primavera de Filippi
Harvard/CNRS
Blockchain lawyer, firestarter, governance researcher

Seth Frey
University of California, Davis
Data scientist, social scientist, expert in online institutions

Nathan Schneider
University of Colorado Boulder
Journalist, democracy activist, platform expert

DAOstack, Blockchain and the Law, Blockchain and the Future of Distributed Governance (ERC grant)

Ostrom Workshop, Computational Institutional Analysis, Data Science of Community Building

Everything for Everyone, Media Enterprise Design Lab, Open Society Foundation Fellow
Team

Joshua Tan
University of Oxford
Computer scientist, mathematician, entrepreneur

Ricardo Saavedra
System designer, HCI researcher, data visualization expert

Louis Kang
MIT Media Lab
Neuroscientist, MMO expert, entrepreneur

Applied Category Theory, Compositionality, Local Sense Lab

XAI, Petal, Sybl

Connectome, SavvyStat
Industry partners committed to change
Seed

Seed is a massively-multiplayer simulation game being developed by Klang Games. Our core team has been working with them for over three years, helping them design the political and economic systems of the game.

“We want players in Seed to experience a world with a huge range of different kinds of politics. We want to see constitutional democracies, medieval monarchies, corporate dictatorships, and 70s-style communes all emerging out of player-to-player interactions. We are incredibly excited to be working with Larry, Josh, and the Metagov team to make this possible.”

Mundi Vondi
CEO of Klang Games

A massive, dynamic world

Traditional sandbox MMOs like Minecraft allow players to build vast, empty cities that have no purpose except to be seen. In Seed, millions of players will come together to build vibrant cities, companies, and even nations.
The perfect testbed for governance

Players on the Seed Discord are already forming factions and imagining their own constitutions. Factions will need to motivate and control their players. Governance is not an option. But governance is also part of the fun.
Vingle

Vingle is an interest-based social network of 5 million users. We are working with Vingle and the CAN Foundation to build more robust self-governance tools for their interest communities.

“At Vingle, we want to explore a new paradigm of collective decision-making. We’re very excited to be working with the Metagovernance Project on a governance designer to support our user communities.”

Changseong Ho
President, COO
Roadmap overview

- **2020**
  - Proof of concept deployed with one of our partners
  - Modules/editor research & development (user interviews, workshops, prototypes)

- **2021**
  - Expand partnerships to incentivize protocol adoption (games, blockchains, etc)

- Scalable module/agreement engine

**Sustainable funding strategies** (e.g., open-source + software as service for premium features)
1-year roadmap

Phase one
Feb | Mar | Apr | May | Jun | Jul | Aug | Phase two
Sep | Oct | Nov | Dec | Jan

Initial Prototypes
- Metagov v0.1: Govlist
  - Govlist roll-out, iterate
- Constitution Web App
  - App roll-out, iterate
  - Iterate software spec, user stories
- Theoretical framework
- Metagov v0.3: Parallel Prototypes

Deployable Prototypes
- Metagov v0.4: Agreement Engine
  - Core module development
  - Mod developer community management
- Metagov v0.5: Module Design Language

Deployed Proof of Concept
- Partner integration
- Partner Workshops
- Alpha Testing
  - Reflect on Outcomes

Metagov
- PM: Project Management
- RT: Research Team
- DEV: Development
- ERT: Extended Research Team
- ALL: All
Key challenges

Translating science into engineering

Metagov is both a basic research project as well as a software development project that will be deployed on live, commercial platforms. The challenge is to balance the science and the engineering. Mapping research questions to engineering requirements is important, as is forming long-term collaborations with external researchers. The core team is experienced in running such research-backed software ventures, and we are very confident that we can address this risk.

Platform adoption

Not all platforms are interested in enabling self-governance or decentralization. We will get around this by seeking out ideological partners, building our own connectors to certain platforms, and by conducting focused advocacy that argues for the legal and economic value of self-governance.

User adoption

Governance is necessary, but it can get complicated. Once embedded in a platform, the toolset needs to be easy to use. We will take a prototype-and-iterate approach to make sure that whatever we build will appeal to actual users.
Vision

A governance layer for the Internet. A sandbox where we can test and implement not just software but also laws, norms, and markets. Simple building blocks (modules) built on a rigorous foundation (computational agreements), but leading to a huge possible range of editable governance forms and purposes, from traditional democracies to nonprofit corporations to minimalist libertarian communities. Scientific tools to study, analyze, and edit the behavior of these governance systems, allowing feedback to users and robust experiments. And finally, tools to scale up the successes and prune the failures.

“If the digital future is to be our home, then it is we who must make it so.” – Shoshana Zuboff
Vision

Too many would-be platforms regard governance as a secondary activity. Too many would-be tools think of governance in simplistic terms, in terms of proposal systems and majority voting schemes.

As communities mature, users’ wants and behaviors and expectations change, transforming the platforms that host them. But without effective governance, communities cannot mature.

Without governance, a game will always be just a game. With governance, it can become something more.
part of the Metagovernance Project

www.metagov.org
Research Overview
Metagov is a metaconstitutional tool. It is designed to create effective constitutional systems (a.k.a. user groups). Users then define the rules and laws of collective action within these constitutional systems.
From research questions to software requirements

Questions

How do we facilitate innovation in (self-) governance?  
What are the building blocks of governance?  
How portable is governance?

Requirements

Modularity: users need to be able to vary governance mechanisms (including code) easily and predictably.  
Interoperability: modules A, B developed for platform X work together, enabling them to function together as a coherent system.  
Portability: modules (e.g. election systems) developed for platform X also work for platform Y.
The lifecycle of governance

Stage 0
- Transition Trigger
- State of Nature
  - External forums
  - External chatting
  - Messaging
- Deliberation Space
  - Proposal
  - Forum
    - Proposal management (e.g., upvoting or
      holographic consensus)
    - Document editing
- Choose (One, Few, Many) and Modify

Stage 0.5
- Agreement Engine
  - Agreement Registry
    - Law Archive
      - Constitution
        - Police, Courts, Reputation, Code
      - Enforcement
        - Record of Infractions and Punishments

Stage 1
- Stage 2
- Stage 3
- Revolution Trigger

Metagov
### The ontology of governance structures

An open database and API that organizes and classifies 150+ governance structures. Built to support later work on module and protocol design. [metagov.org/govlist](https://metagov.org/govlist)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Description</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td><strong>Level</strong> Institution or concept</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74</td>
<td></td>
<td><strong>Module</strong> Association (unincorporated)</td>
<td>Unincorporated associations are generally formed by the action of a number of individuals in associating themselves together under a common name for the accomplishment of some goal.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td></td>
<td><strong>Module</strong> Smart contract</td>
<td>A smart contract is a computer protocol intended to digitally</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td></td>
<td></td>
<td>facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. These transactions are trackable and irreversible.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77</td>
<td></td>
<td><strong>Module</strong> Simple contract</td>
<td>A smart contract in which most of the logic is run-offchain, with only the ability to hold funds and release funds based on voting remaining on-chain.</td>
<td><a href="https://medium.com/@ryanshea/simple-contracts-are-better-contracts-what-we-can-learn-from-the-dao-6293214b4ad3a">https://medium.com/@ryanshea/simple-contracts-are-better-contracts-what-we-can-learn-from-the-dao-6293214b4ad3a</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td></td>
<td><strong>Module</strong> Permission system</td>
<td>Hierarchical scheme for assigning rights over a set of resources to different types of agent. Agents can be of many types. A bundle of rights that determine which actors have been authorized to perform which actions with respect to a specified resource.</td>
<td><a href="https://en.wikipedia.org/wiki/Wikipedia:User_access_levels">ACL, https://en.wikipedia.org/wiki/Wikipedia:User_access_levels</a></td>
<td>resource allocation</td>
<td></td>
</tr>
</tbody>
</table>
Modular Politics

A journal article which describes some of the key building blocks of online self-governance, including modules and orgs (a.k.a. agreements), and imagines several examples.

metagov.org/modpol
The Metagame

Design language of governance

A research project to construct:

1. A computational representation of social rule systems
2. A design language that enable users to combine modules
Design Appendix
Module View

The module view organizes modules according how they fit together. From left to right, modules are ordered according to their relationship to one another.

Non-technical users can use pre-made modules to create organizational forms, while technically-inclined users can go underneath the hood to edit the modules according to their own needs.
Usability vs Abstraction

Visual programming languages are quite powerful and easier to use than command line programming. However, we believe that modules should exist at a higher level of abstraction, enabling users to think creatively with modules and how to combine them to create different structures.

We believe that a mix of simple to use modules (already pre-configured) with more granular and computationally driven node-based programming can exist within metagov.

We want users to be thinking about governance, not programming.
Interaction Model References

The project below (programmable blocks) strikes a good balance in making things feel simple. We're using a similar approach in our design language. We want our interface to look and play like a game, rather than something stiff and serious.
Interaction Model

Modules and their rules are summarized in a high level "constitution" format from which everything is made transparent.

Ready-made modules are easily available and ready to be configured.

Node-based language from which new modules can be assembled from.
Modules Examples

The list below is just an example of a few module we're considering implementing first. Each module will have editable properties and ways to customize its behavior.

- **Voting**
  - Lottery
  - Approval
  - Condorcet
  - Quadratic
  - Continuous

- **Curation**
  - Staking
  - Token curated registry

- **Resource Management**
  - Automated Payments
  - Tax collection

- **User Group**
  - Board Member
  - User Groups
  - Federations

- **UI**
  - Sliders
  - Radio buttons

**Properties**

- Delegation
- Election Period
- Eligible voters
- Vote weight
Module notification: if new interaction is required from a user, it will show up within a specific module. A user will need to click and perform an action in order to de-activate it. Actions are: voting, delegation, and so on.

Module Properties: edit all the available properties for a module. (e.g. by instantiating a module, you have to set its parameters in order to instantiate it. There are no defaults for modules, users need to set them up)

Module UI: is where you might go as a user to interact with a parameter (e.g. a voting module comes with a voting UI that gets created after all the properties are defined)
Constitution

Preamble
Our main goal is to make players wealthy, especially newcomers. We want people associated with us to be the richest players in the game. We wish to provide advantageous contract deals for Business Owners, Investors, Employees, as well as teaching them a guide on how to achieve financial success.

Proposal Submission
All proposals will be voted on using a quadratic voting system.
1. import members as @
2. import voting, quadratic_voting
3. members_list = [@Josh, @Nathan, @Primavera, @Louis, @Larry]
4. voting.mechanism = [voting]
5. voting.options = quadratic
6. if voting.votes > 10:
7.   voting.vote_outcome = exception.invalid

Quadratic Voting
1. quadratic_voting.options = (k,1,d)
2. quadratic_voting.period = (5, days)
3. quadratic_voting.scaling = 2.1

Roles
All roles will be filled through direct elections.
1. import roles
2. roles.executive = @Dob
3. roles.representation = @Nathan, @Primavera, @Seth
4. if roles.executive = empty:
5.   voting.succession

Amendments
1. import amendments
2. amendments("Proposal Submission") = [greater than .5, all_members]
3. amendments("Roles") = fixed

Quadratic Voting

Properties

Vote Eligibility

Include Members

Include Members

Voting Period
3 days

Voting Period
3 days

Voting Period
3 days

Interface

A voting period is open: please select 3 candidates from all users from which you'd like to support. If you're not sure how to proceed, you can review all the rules set for this module, and read more about Quadratic Voting in our wiki. If you're still not sure, you can always contact the administrator for support.

Select Candidates

Susan M
Add Candidate

Added Candidates

Cullem Miller  Catrom Mauere  Ostrom Mauere

Confirm Changes
Monitors

As a (partial) fork of the Jupyter notebook, Metagov comes with native support for embedded data visualizations. Users can access all the data within Metagov and plot it using pre-made data visualization modules (e.g. who voted for whom in a past election, which proposals are accepted and who voted on them). This allows continuous feedback to governance systems.