**Waiting For You项目介绍**

**目录**

1 项目愿景 1

2 项目设计 3

基本法设计 3

领域应用场景设计 4

项目设计背景 4

项目设计理念 4

业务逻辑 5

项目架构 6

3 项目实施 7

业务流程 7

4 开发进度 9

智能合约 9

测试版界面展示 17

# 1 项目愿景

构建一套社区规则治理体系。

**目前治理体系的重大弊端：**

1. 谋事不如谋人
2. 腐败不可避免
3. 管理弱于政治
4. 利益大于良知
5. 民众幸福感消失

**造成目前治理体系重大弊端的原因：**

1. 资本控制一切，对资本的追求永无止境
2. 用人机制落后，以人用人导致利益输送
3. 资本权利过大，拥有金钱等于拥有一切
4. 权利寻租，以权换钱的驱动力极强
5. 资本向少数人集中，多数人的基本权益被合法剥夺

**这些原因的本质是由于构建社会法则的基础出现了问题：**

以资本为单核建立的社会框架和法律体系决定了整个社会都不断追求资本，而资本又是趋利的，导致资本追逐永无止境的同时，资本愈发地集中，大多数人的财富合理合法地转移到少数人的手中。

共产主义的物质极大丰富是一个伪命题，因为只注重客观物质的唯物主义者忽视了唯心层面的因素。人的幸福感来源于内心而不源于外部，物质永远不可能极大丰富，因为人的贪欲是无穷的，再多的物质也满足不了贪婪。同时，在无产者取得革命胜利后，没有能够将权利和资产分开设计，重新走向了资本为单核的道路，导致党内基本框架形同虚设，上下管理者矛盾不可避免。在资本为单核的挟持下，社会治理走向恶性循环，人和人的关系滑入囚徒困境，无法逆转。

**解决问题的思路：**

资本体系不能完全否定，货币作为社会资源交换的媒介具有无可替代的作用。利用货币手段有助于快速集中资源力量进行庞大项目的建设。适当的趋利性能够提高社会人的生产动力等。

资本的力量必须进行限制，放纵资本将带来恶果。资本必须置于机构的管辖之下，同时，机构必须游离于资本之外，才能确保机构不被腐蚀。机构是由人构成的，所以机构的成员选拔不能由人来约束，而应该由法则来规定。但是，由于法则归根结底是为了进行社会管理，而法则的制定除了由超然力（例如自然规则）来制定以外，只能由人类自己制定，故而法则必须由最广大的民众制定才有公约力。一旦法则不受公约的限制而由某个机构进行制定，那么整个社会必然又会落入到少数人压榨多数人的恶性循环当中。

**解决问题的办法：**

必须设计一套独立于资本的权力体系，必须保证权力最终由最广大的用户赋予，这是项目设计的基本法则。

整个社区治理的基本要素有：

1. 用户
2. 管理者（协调者）
3. 圣裁院

其中，用户是整个社区的参与者，是整个社区财富和资源产生的来源。管理者是社区资源的调度者、社会治理规则的建议者，拥有很多社区财富的支配权、社区治理规则的建议权、区域治理的决定权。裁决者是整个社区规则的制定者、社区环境的建设者、社区软硬件的维护者。

社区必须建立由区块链智能合约为基础的基本法。由于区块链具有不可篡改与公开的特性，任何一个人都可以通过唯一的id去审视基本法，而后去决定是否参与到社区当中，成为社区用户。由于没有用户便没有社区，因此基本法必须能够保证用户的利益才会被接受。当拥有用户后，社区便形成，此时开发团队的身份转换为圣裁院。

圣裁院拥有很大的裁决权限，可以根据实际需要完善社区体系。但是，圣裁院的一切行为受到基本法的约束。圣裁院所做的一切工作需要抵押荣誉通证，当这部分工作被广大用户认可后，其荣誉通证会根据用户的认可度按抵押比例增加；当不被认可时，则按抵押比例扣除。荣誉通证是进入圣裁院的门槛，意味着圣裁院成员不是一成不变的。申请者必须抵押足量的荣誉通证。当满足标准以后，其成员由选举产生，选举人为所有用户，但为了避免泛民主化带来的恶果，选举的票并非是一人一票，而是可以通过抵押荣誉通证进行选举。用户必须持有一定的荣誉通证才拥有选举权，而其抵押的荣誉通证必须不超过拥有选举权的最低荣誉通证的多出的部分，以此来确保专业者和道德者的话语权。抵押荣誉意味着用户将与自己所选的人员荣辱与共，当尤其支持的圣裁政策不得民心时，所有支持者抵押的荣誉通证将被按比例扣除。如此一来，选举人必须选举对整个社区有益的候选者才能保证自身权益，避免了选票绑架。

管理者的产生与圣裁院类似，管理者的量化同样需在基本法的框架下，由创世管理者提交该领域管理者的量化提案，圣裁院必须审核提案的公正合理性，通过后向创世管理者提供相应的资源。管理者量化提案一旦通过，即作为领域的基本法，不可更改。

社区基本法的制定需要遵循的原则为：必须以用户作为唯一评判标准。其分为，参与社区用户的人数，以及用户对政策的支持或反对情况。由于用户选举需抵押自身荣誉通证，因此绝大多数用户将不会随便对管理者或圣裁院的政策进行反对。因此，一旦用户开始流失或者有大量的反对声音出现，那么被反对者将被默认为政策失误，从而扣除掉其抵押的荣誉通证。



1.1 社区治理的基本格局

# 2 项目设计

为实现项目愿景，需要对项目进行设计，并分步实施。

## 基本法设计：

基本法是项目构建所必须遵循的法律，基本法必须简明，逻辑必须完整合理。基本法的组成为：

1. 双通证：项目发行资产和荣誉双通证，其中，资产通证是资产价值交换的媒介，荣誉通证决定个人在社区中拥有的权益。
2. 圣裁院：圣裁院每年换届一次，圣裁院成员申请者所拥有的荣誉通证量必须位于所有社区用户的前10%。
3. 民众监督：圣裁院的工作必须通过智能合约完成，智能合约必须开通民众提交支持或反对态度的接口。圣裁院提交合约必须经过半数以上成员签名，并抵押不少于最低要求的荣誉通证才能生效。当民众反对人数超过总人数的1/3时，将根据反对的人数按比例扣除圣裁院签名成员的荣誉通证；当民众表示支持的人数超过总人数的2/3时，将根据支持的人数按比例增加圣裁院签名成员的荣誉通证。当圣裁院任职期间用户流失超过一定水平时，将根据流失人数按比例扣除圣裁院全体成员的荣誉通证。
4. 选举：所有持有最低标准量M以上的荣誉通证的用户均拥有选举权和被选举权。其权利在一次换届中不能同时行使，即不能够在一次换届中既是选举人又是被选举人。
5. 选票：每个社区用户的选票即为荣誉通证的抵押。可抵押的通证量为持有量X减去保证选举权的最低标准量M。
6. 交易：资产通证可交易，交易为账户的一种权益。账户每日可交易额度与账户荣誉通证的持有量成正比，交易手续费与荣誉通证的持有量成反比。
7. 现实连接：所有需要进行现实连接的账户必须提供现实身份信息。没有经过现实连接的用户仅能使用社区提供的虚拟世界功能。

## 领域应用场景设计：

基于基本法，为使项目落地，需要进行领域应用场景设计。

Waiting For You项目是基于社区基本法的第一个领域应用项目。该项目旨在搭建一个更加高效的寻人启事平台，帮助失散者和寻人者尽快团聚。Waiting for you的服务目标群体主要分为两类，一类是寻人者与失散者有共同的团聚意愿，且失散者有完全独立的民事行为能力；另一类是寻人者有团聚意愿，失散者无独立的民事行为能力。任何有完全独立民事行为能力的一方没有团聚意愿，则不在该平台的服务范围之内。

## 项目设计背景

全国每年失踪人口24万，全国每年寻人人口大于200万。我国现有官方寻人平台一个：公安部儿童失踪信息紧急发布平台。2016年5月15日开放至今找回失踪人口3100人。我国民间寻人平台超过100个，总发布信息超过30万条，找回率不足1‰。

造成失踪人口找回数量与比例都很低的原因主要有以下三个：

1、寻人机构成本与效率之间不可调和的矛盾。

2、激励措施单一，尤其是道德激励力度不足，付出向成效转化不明显，参与者获得感不强。

3、大规模发动群众的推动力不够，群众参与度不高。

Waiting For You则是解决以上三个问题的寻人平台。

## 项目设计理念

为解决目前寻人手段面临的各类问题，Waiting for you充分利用区块链技术带来的分布式自治特性，利用通证生态为全民参与提供内生动力。

1）双通证

社会活动是人类行为体系中典型的集体行动,它必然会在行动中突出甚至公开宣示自己的价值。同样的，在寻人过程中，参与的各方均有显示自身价值的意愿。但是，在目前以货币作为价值衡量唯一标的的条件下，道德的地位下降严重。寻人由于重在产生道德价值，产生的经济价值很少，导致即使是政府机构也不愿过多参与到寻人活动当中。因此，为了解决目前这一问题，Waiting for you采用资产与道德双通证策略。资产通证用于满足资产交换的需求，道德通证用于提供权益获得与资源支配的衡量标准。道德通证的持有量将决定个人在平台中拥有的权益，从而避免财富支配一切的弊端。

2）双向寻人

寻人的最优方式是能够充分发动各地的群众参与其中。但是，由于人对自己陌生的相貌难以形成较深的印象，故在绝大多数情况下，即便群众有意愿帮助寻人，但由于记不住失散者的相貌，导致即使确实在生活中遇到了失散者，也会因没有印象而与其失之交臂，不能有效地转化为线索提供出来。

为弥补这一不足，Waiting for you设计双向寻人功能。在已有寻人手段的基础上，添加失散者和路人随机线索上传功能，鼓励群众合规发布须被救助者的信息，寻人者通过查阅发布的信息，寻找失散者的线索。由于寻人者对失散者的印象比较深刻，故当线索被发现时，寻人者能够立即作出判断并加以采用，从而提高了寻人效率。

3）治理体系

Waiting for you不由某个特定的人或机构治理，管理者完全根据链上规则竞争选出。将道德通证作为参与平台治理的准入证明，通证的获取、管理者的任免等由智能合约进行评价量化，而具体治理则完全由管理者把握，从而形成智能合约量化——道德者治理——公众参与的去中心化治理体系，确保平台的稳定扁平化运行。

## 业务逻辑

Waiting for you的寻人业务分为两类，第一类为优化的常规寻人业务，寻人者根据自身需求发布寻人信息到寻人栏，同时设置寻人奖金到赏金池；投票者根据对寻人信息的关注度，视情向赏金池中投票以追加赏金，提升寻人信息的重要度；评论者根据掌握的线索提供信息以获得寻人奖金；寻人者发布的信息以及评论者发布的评论需经审核维护者进行审核，提出并不断补充价值评论名单。当实现寻人目的时，由寻人者和投票者共同对价值评论名单进行确认，当存在异议时，需在规定时间内提出替代名单并获半数以上支持。名单确认后，系统按照名单将赏金池中的票按比例分配给价值评论者和审核维护者。其业务逻辑如图2.1-a所示：

2.1-a 寻人者发布信息时的业务逻辑

Waiting for you第二类寻人业务，由失散者或热心人将失散者信息和线索发布到寻人栏，投票者根据对寻人信息的关注度，对线索进行投票以提升线索关注度；寻人者根据自身需求浏览寻人栏中失散者的信息，发现并获取价值线索提供者的联系方式。发布的信息同样需经审核维护者进行审核。当实现寻人目的时，线索提供者和投票者都将获得相应的奖励。其业务逻辑如图2-b所示：



2.1-b 失散者或热心街拍者发布信息时的业务逻辑

## 项目架构

Waiting for you采用三层架构：共识层、存储层与交互层。共识层利用EOS.IO底层技术，用于管理平台功能、社区治理规则体系、栏目分类、文章目录及摘要；存储层利用星际文件系统对用户文件详细内容、社区法律文件、统计数据；交互层为用户提供与平台的交互接口，使用户方便地使用平台各项功能。



2.2 失散者或热心街拍者发布信息时的业务逻辑

1）共识层：Waiting for you通过智能合约实现平台功能，制定社区治理规则体系，并记录平台栏目、文章创建情况，所有平台行为将被忠实记录并公开，便于明确权责并对争议行为进行追溯判定。

2）存储层：利用星际文件存储技术（IPFS），Waiting for you可以高效地记录各类文件数据，便于在任何地点进行快速的存取。同时，IPFS提供的文件哈希非常便于区块链系统的存储，降低了平台的数据存储成本，提升了平台数据存储的安全性。

3）交互层：Waiting for you通过对EOS.IO接口进行封装，建立一个面向用户的UI操作端。用户可以通过操作界面轻松实现平台上的各类操作。

# 3 项目实施

为确保项目的顺利实施，我们首先设计并开始构建寻人启事平台的1.0版本。

## 业务流程

在1.0版本中，票（MZP）由代币（MZ）抵押获得，项目中存在审核维护者、投票者、寻人者、评论者四个角色。寻人者根据自身需求发布寻人信息到寻人栏，同时设置寻人奖金到赏金池；投票者根据对寻人信息的关注度，视情向赏金池中投票以追加赏金，提升寻人信息的重要度；评论者根据掌握的线索提供信息以获得寻人奖金；寻人者发布的信息以及评论者发布的评论需经审核维护者进行审核，提出并不断补充价值评论名单。当实现寻人目的时，由寻人者和投票者共同对价值评论名单进行确认，当存在异议时，需在规定时间内提出替代名单并获半数以上支持。名单确认后，系统按照名单将赏金池中的票按比例分配给价值评论者和审核维护者。其业务流程如图3.1：



图3.1 寻人启事1.0版本业务流程

# 4 开发进度

项目采用eos+ipfs+react框架进行开发，目前开发进度如下：

## 智能合约

Waiting for you的智能合约基于EOS技术，利用李嘉图合约对合约进行法律性限制。智能合约分为基础合约及版本合约。基础合约即平台基础法，规定平台治理的基础框架，包含双通证合约、圣裁合约等，不可修改。版本合约是对平台功能及治理模式的改进，可在基础法允许的范围内进行更新迭代。

##wafyarttoken合约接口

1、创建代币，用户不可以调用，只能合约内部调用。

void create(account\_name issuer,asset maximum\_supply);

//参数：

//issuer：发行代币所属账户。

//maximum\_supply:代币发行的最大数目

//签名：

//由wafyarttoken账户进行签名调用

2、发行代币，用户不可以调用，只能发行代币所属账户调用。void issue(account\_name to, asset quantity, string memo);

//参数：

//to：将代币发行到的目标账户。项目中此账户为wafyarttoken

//quantity：发行的代币结构asset，包含数量和符号，

//memo:备注，不超过256字节

//签名：

//由wafyarttoken账户进行签名调用

3、转移代币，用户可以调用。

void transfer(account\_name from,account\_name to,asset quantity,string memo);

//参数：

//from:token转移的源账户，即为eos区块链的账户名。(不超过12位，取值为a-z1-5)

//to:类型同from，token转移的目标账户

//quantity: asset类型，例如"1234.0000 MZ"

//memo:备注，不超过256字节

//签名：

//由from账户进行签名调用

4、抵押代币，用户可以调用，MZ代币抵押后成为MZP

void staketoken(account\_name byname,asset quantity);

//参数：

//byname:操作的账户

//quantity:asset类型，抵押的token数量

//签名：

//由wafyarttoken账户进行签名调用

5、增发代币，用户奖励价值评论者，由合约内部调用，用户没有权限调用，不用测试

void addtoken(account\_name byname,uint64\_t amount);

//参数：

//byname:操作的账户,在项目中只能为wafyartvotes合约

//amount:增发的token数量

//签名：

//由wafyartvotes账户进行签名调用

## wafyartvotes 接口

1、抵押代币，用户不能调用，由wafyarttoken的staketoken方法调用，不用测试

void staketit (account\_name byname,account\_name accname,uint64\_t amount);

//参数：

//byname:签名者账户，项目中设置为wafyarttoken账户

//accname:抵押代币的账户

//amount:抵押的代币数目

//签名：

//由wafyarttoken账户进行签名调用

2、解锁抵押的代币,MZP==>MZ，比例1：1，用户调用，3天之后解锁

void unstaketit(account\_name byname,uint64\_t amount);

//参数：

//byname:解锁代币的账户

//amount:解锁的代币数目（为了保持精度，1MZP的数目为10000。即解锁10MZP，amount为100000）

//签名：

//由byname账户进行签名调用

3、 设置解锁记录状态，用户不可调用，不用测试，内部调用。

void setstats (account\_name byname,account\_name accname,uint64\_t id)

//参数：

//byname:wafyartvotes合约

//accname:操作的账户

//id:解锁代币记录id

//签名:

//由wafyartvotes合约内部调用

4、 撤销解锁，用户调用，在解锁过程中是可以撤销操作的。

void delunstake(account\_name byname,uint64\_t id);

//参数:

//byname:需要解锁代币的账户

//id:解锁代币记录id，通过accunstakes table获取到

//签名:

//由byname账户签名调用

5、给文章投票，用户调用，投票之后，会根据投票率，增发奖励给最佳评论者，投票为3天后恢复

void voteart (account\_name byname,uint64\_t votenum,account\_name catename,uint64\_t artid);

//参数:

//byname:投票的账户

//votenum:投票的数目（为了保持精度，1MZP的数目为10000。即投票10MZP，votenum为100000）

//catename:分类名，类型同账户名(a-z1-5,不超过12位)

//artid:文章id，由articles table获取

//签名:

//由byname账户签名调用

6、赎回投票，由合约内部调用，不用测试

void redeemvote(account\_name byname,account\_name accname,account\_name catename,uint64\_t amount);

//参数:

//byname:wafyartvotes账户

//accname:该赎回投票的账户

//catename:分类名

//amount:该赎回的票数

//签名:

//由wafyartvotes账户调用

7、撤销文章的得票数，由合约内部调用，不用测试

void redeemart (account\_name byname,uint64\_t artid,account\_name catename,uint64\_t amount);

//参数:

//byname:wafyartvotes账户

//artid:文章id

//catename:分类名

//amount:撤销的票数

//签名:

//由wafyartvotes账户调用

8、撤销审核员的得票数，由合约内部调用，不用测试

void redeemaud (account\_name byname,account\_name audname,account\_name catename,uint64\_t amount);

//参数:

//byname:wafyartvotes账户

//audname:审核者账户

//catename:分类账户

//amount:撤销的票数

//签名:

//由wafyartvotes账户调用

9、申请成为某个分类的审核者，由用户调用，成为审核者需要消耗100MZP

void regauditor(account\_name byname,account\_name catename);

//参数:

//byname:申请者账户

//catename:分类名

//签名：

//由byname账户签名调用

10、删除自身审核者身份，消耗的100MZP不退还

void delauditor(account\_name byname,account\_name catename);

//参数:

//byname:删除者账户

//catename:分类名

//签名:

//由byname账户签名调用

11、给审核者投票，投票分类下的审核者和投票分类下的文章，获取的收益相同。

void voteaud (account\_name byname,uint64\_t votenum,account\_name catename,account\_name auditor);

//参数:

//byname:投票者账户

//votenum:投票的数目（为了保持精度，1MZP的数目为10000。即投票10MZP，votenum为100000）

//catename:分类名

//auditor:审核者账户

//签名:

//由byname账户签名调用

12、创建文章，创建文章需要消耗MZP，消耗的数目最小值有创建分类时定义

void createart (account\_name byname,string title,string abstract,ipfshash\_t arthash,account\_name catename,uint64\_t payticket);

//参数:

//byname:文章作者

//title:标题，不超过40个字节

//abstract:摘要，不超过400个字节

//arthash:文章内容的ipfs哈希值

//catename:文章所属分类名

//payticket:支付的MZP数目

//签名:

//由byname账户签名调用

13、修改文章，标题、摘要、文章内容均可修改

void modifyart (account\_name byname,string title,string abstract,uint64\_t id,account\_name catename,ipfshash\_t newarthash);

//参数:

//byname:文章修改者

//title:新标题

//abstract:新摘要

//id:要修改的文章id

//catename:文章所属分类名

//newarthash:新文章ipfs 哈希值

//签名:

//由byname账户签名调用

14、删除文章，可以由作者本人删除，也可以由审核员删除

void deleteart (account\_name byname,uint64\_t id,account\_name catename);

//参数:

//byname:删除文章的账户

//id:文章id

//catename:文章所属分类

//签名:

//由byname账户签名调用

15、创建分类，文章都是基于分类的，投票奖励也是基于分类的，消耗500MZP

void createcate(account\_name byname,account\_name catename,string memo,uint64\_t paylimit,uint64\_t auditnum,uint64\_t comratio ,uint64\_t voteratio, uint64\_t audiratio, uint64\_t devratio);

//参数:

//byname:分类创建者

//catename:要创建的分类名称

//memo:分类介绍，不超过400字节

//paylimit:该分类下，发布文章，需要消耗的最小MZP数目(为了保持精度，1MZP的数目为10000。即想设置10MZP，paylimit为100000）

//auditnum:审核员数目,审核员数目没有满员，该分类下无法发布文章，范围：大于1小于100

//comratio:价值评论获得的奖励的分成比例,大于1，小于100，奖励是指发布文章消耗的MZP

//voteratio:投票者获得的奖励的分成比例，大于1，小于100

//audiratio:审核者获得的奖励的分成比例，大于1，小于100

//devratio:开发资金、资源购买获得的奖励的分成比例，大于1，小于100。comratio+voteratio+audiratio+devratio=100

//签名:

//由byname账户签名调用

16、创建订阅

void createscr (account\_name byname,account\_name catename);

//参数:

//byname:订阅者账户

//catename:分类名

//签名:

//由byname账户签名调用

17、删除订阅

void deletescr (account\_name byname,account\_name catename);

//参数:

//byname:删除订阅者账户

//catename:删除的分类名

//签名:

//由byname账户签名调用

18、创建评论，评论分为1级评论和2级评论

void createcom (account\_name byname,string comcontent,account\_name catename,uint64\_t parid,uint16\_t indexnum);

//参数:

//byname:评论创建者账户

//comcontent:评论内容，不超过400字节

//catename:评论所属分类名

//parid:父id，一级评论父id为文章id，二级评论父id为一级评论id

//indexnum:标志，为1或者2，表示一级评论或者二级评论

//签名:

//由账户byname签名调用

19、删除评论，删除一级评论，会包括下属二级评论一起删除。删除二级评论，只删除该评论自身

void deletecom (account\_name byname,uint64\_t id,account\_name catename,uint16\_t indexnum);

//参数:

//byname:删除评论者账户

//id:评论id

//catename:评论所属分类名

//indexnum:评论等级，为1或者2

//签名:

//由byname账户签名调用

20、设置价值评论，价值评论得主会获得奖励，奖励包括增发的部分

void setbestcom(account\_name byname,uint64\_t artid,uint64\_t comid,account\_name catename);

//参数:

//byname:操作者账户

//artid:文章id

//comid:价值评论id

//catename:所属分类名

//签名:

//由byname账户签名调用

21、设置文章锁定期，锁定期是指文章发布7天之后，不允许投票。合约内部调用，不用测试

void setartend (account\_name byname,uint64\_t artid,account\_name catename);

//参数:

//byname:wafyartvotes账户

//artid:文章id

//catename:文章所属分类名

//签名:

//由账户wafyartvotes账户调用

测试版界面展示









