



Senno 白皮书  
v1.0.2

[Senno.io](https://Senno.io)



我曾有过一个关于网络的梦想，这个梦想共有两个部分.....在第二部分里，网络可以与电脑联通。机器可以在网络上分析人与电脑之间的数据，包括内容，关联和交易等。要实现这种可能，一个所谓的“网络”就必须先出现，然而，当它出现之后，关于贸易、官僚主义和我们的日常生活的机制都将交给机器与机器的交流来实现。这个人们期待已久的智能“代理”终将出现。

*Tim Berners-Lee, Weaving the Web, 1999*



Σ

Senno是第一个为第三方App提供API接口的情感分析平台

它利用分布式情感分析和先进的人工智能算法打造了一个实时的群众智慧生态系统和复杂的商业智慧分析。

Senno将为商业和私人领域的决策方式带来一场变革。



<b>引言</b>	<b>4</b>
用户生成内容大量涌现	4
需求和解决方案	5
Senno网络使用流量	6
<b>Senno的主要特点</b>	<b>7</b>
<b>NEO - Senno区块链</b>	<b>9</b>
为什么选择Neo ?	9
<b>Senno背后的数学</b>	<b>14</b>
词汇分类	14
寻找贡献者	15
Senno输出和提供的信息	17
<b>Senno代币 ( SENNO )</b>	<b>19</b>
用途和目的	19
存款和提现	19
The SENNO 代币合约	20
Senno客户端	21
<b>认可贡献</b>	<b>22</b>
硬件贡献奖励	22
可持续经济	23
平台的扩展	23
衍生和定制版本	23
加盟和合作关系	24
<b>金融情绪</b>	<b>25</b>
商业模式和货币化	25
将区块链和加密货币带给大众	26
CryptoScanner - Senno的第一个应用程序	26
CryptoScanner的功能	27
参考实现	28
<b>发展规划</b>	<b>29</b>
<b>SENNO的代币销售活动</b>	<b>30</b>
代币销售经济	31
资金使用	32
<b>Senno团队</b>	<b>33</b>
顾问团队	34
合作伙伴	34
风险因素	35
监管策略	36
参考文献	37
Code implementations	38
Disclaimer	40

# 引言

在过去的十年中，Web 2.0网站变得越来越流行，进行互相交流和合作的用户一直呈现指数增长。用户生成的内容被广泛应用，如解决问题，新闻，娱乐，广告，八卦和研究等。虽然很多公司都明白，他们不能忽视与其品牌或产品相关的用户生成的内容（如社交媒体上发布的帖子），但很少有人将此内容视为理解消费者思维的方式，也没有在日常业务决策中贯彻大众知识。

用户生成内容的神奇之处在于对公众舆论和生活习惯的前所未有的反映。阅读别人写的内容可以窥探他们的生活方式，包括他们居住的地方，他们的样子，他们喜欢吃什么，他们在哪等等。这是因为人们的生活总会与他人产生关联，而不是在毫无逻辑地乱写一通。

通过大量分享用户生成的内容，我们可以分析消费者的真实想法，而不仅仅是像调查中那样仅仅获得了特定问题的答案。了解他们的思维方式，促使他们做出具体决定的因素，他们如何表达自己以及他们使用什么样的言语来描述产品，都能够了解他们对于资产、品牌或产品的看法。

## 用户生成内容大量涌现

互联网的发展创造了一个史无前例的巨大出版空间。任何人都可以分享他们的意见，无论是通过在Facebook或Twitter上的简短评论，还是在Wordpress或Medium上经过深思熟虑的谩骂。

互联网不仅给个人一个表达自己的空间，而且也给予了公司与公众进行互动的机会。从在LinkedIn上发布文章到在他们的网站上共享白皮书，互联网为他们提供了多种新的沟通方式。最重要的是，大量的新闻和其他媒体网站也在互联网创造的新市场上呈现爆炸式增长。

所有这些数字空间已经为情感数据犹如一座金矿，通过复杂的算法和指数增长式的计算能力，便可以对此加以利用。如今，我们可以通过正确的工具和信息来真正理解大众的意愿。这些发展意味着我们终于可以让那些原始的方法，例如根据经验进行推测等不太可靠的方式，成为过去。。



## 需求

第一种情绪分析引擎是复杂的，昂贵的，并且只对少数实体提供。试图找到解决方案的公司遇到了三个主要问题

- 它需要巨大的硬件资源来扫描数百万个网页并实时分析情绪。这意味着运营成本和最终产品都将非常高昂
- 一个不透明的集中心化解决方案存在影响业务可信度的数据操纵风险
- 在Web 2.0的早期，用户生成的内容主要局限于论坛，由早期的采用者制作而成。大多数人没有使用互联网，也没有以有限的方式使用互联网，这意味着普通人群的声音并没有出现在网络上。

尽管在过去的十年中已经出现了各种情感分析技术，但是开发的高成本，以及保存大量数据的服务器设施决定了基于情感分析的产品主要由大企业使用的现实。根据麦肯锡全球研究院的报告，对大数据加以充分利用的零售商能够将营业利润率提升超过60%。

## 解决方案

1

Senno是一个去中心化的，基于区块链的情感分析平台，具有使用分布式硬件的开放式软件开发工具包。Senno允许第三方将群众智慧和数据结论工具整合到自己的平台中（例如，像Alexa和Siri这样的数字助理，基于用户实时提出的关于某个主题或爱好的问题，不管是有趣或无聊，都能够精准地回答）。通过获取公众对某个领域中某个特定实体的指示Senno让公司和个人都能够利用情绪，并让情绪保持紧密相连。

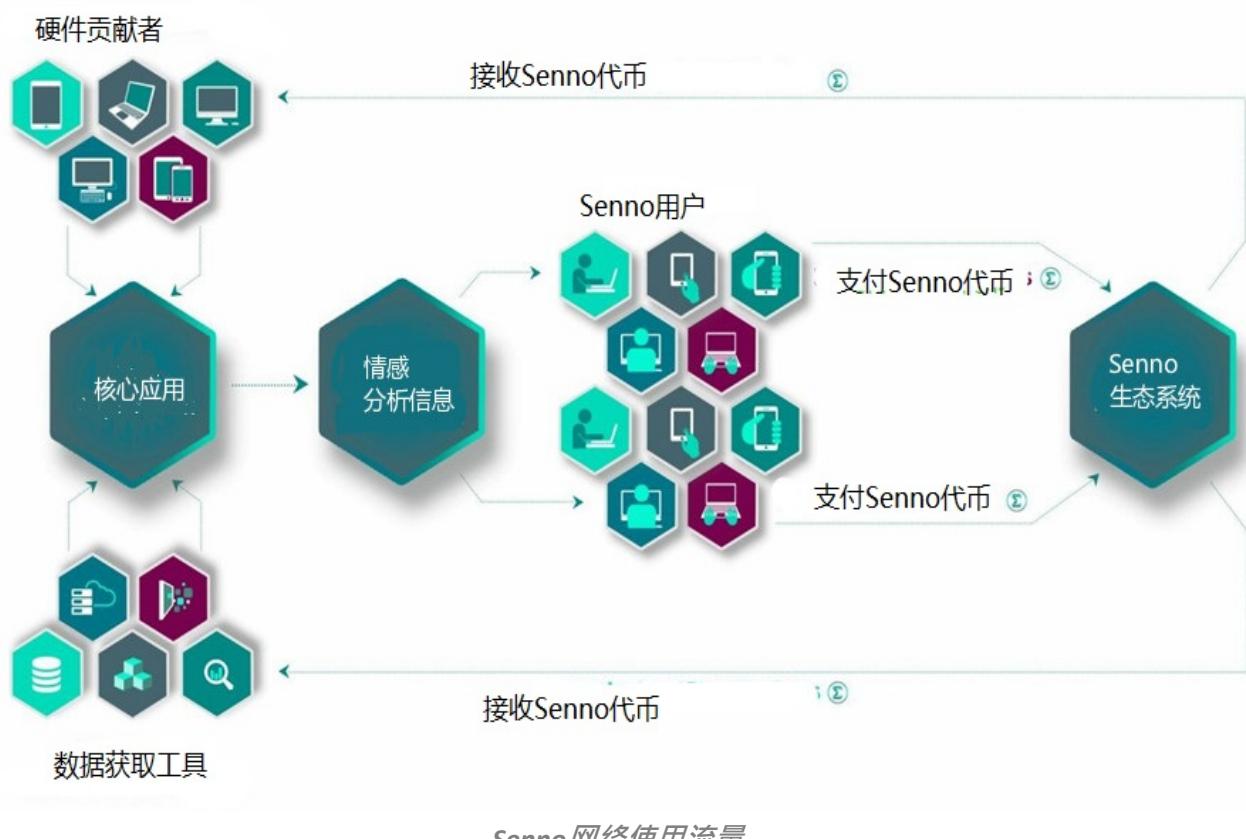
实时接入公众舆论有很多好处，包括能够识别威胁或机会并立即作出反应，以监测营销活动和产品变化的影响，以及特定时间范围或地点的绩效。例如，像公司或加密货币这样的实体的增加可能表明正在发生某些事情，但是不验证它们所代表的内容的情况下进行翻译和采取行动，并分析其中的意义，就显得很困难了。

<sup>1</sup> Big Data The next frontier for innovation, competition, and productivity ([PDF](#))

Senno旨在为第三方提供基础设施和SDK，以准确分析用户情感的实质，并实时生成有价值的信息，然后可以手动或自动作出响应。

通过区块链的智能合约技术，Senno的工程师提出了一个创新的解决方案，可以为所有社区成员提供深入、准确、广泛和实时的情绪数据分析，与传统方案相比，具有费用低，准确性高的优势。

## Senno网络使用流量



# Senno的主要特点

## 贡献奖励体系

2

30%的SENNO代币供应将被锁定在智能合约中，作为贡献奖励。

### • 分布式硬件降低成本

Senno利用其区块链网络的强大功能，允许成员贡献自己的硬件资源，从而消除了维护昂贵数据中心的需求，并且用SENNO代币奖励有贡献的成员。

### • 访问私人数据

使用区块链。系统会使用SENNO代币来奖励提供其私人数据频道访问权限的Senno用户（实时通信APP，微信，QQ，私人新闻频道等）。这些私人数据流通常具有高质量的情感价值，因此，相比起市场上现有的和基于公开可用数据的产品，Senno能更准确地进行分析。

### • 开发贡献

通过为系统开发有特色的功能和插件，社区成员能够获得SENNO代币。由Senno社区票选进入系统的开发模块，将根据开发功能的复杂性和使用情况向开发者颁发SENNO代币。

## 灵活的使用和访问

Senno有一个开源的、去中心化平台，可以通过两种方式进行集成：

### • 开放式SDK

内置的SDK使第三方能够轻易的开发能够利用Senno核心功能的应用程序和插件。应用程序开发人员将能够将情绪分析功能集成到他们开发的软件中，并提供有关其专业领域的提示，信号和分析。

### • API连接

Senno将推出一个开放的API，它将允许Senno分布式存储的快速和动态数据查询。像麦当劳这样的公司将能够使用Senno API来将情绪数据导入他们的BI系统，以测量公众对新产品和营销活动的态度和看法。

<sup>2</sup> See Hardware Contribution Rewarding section (p.22)



## 情绪分析AI

通过运用人工神经网络（ANN），Senno的机器学习算法将能够识别高价值的数据源，并赋予它们更高的内部评分，从而让最终的情感评估具有更高的准确度。此外，该机制还将优化为达到分析所需的数据量。

## 动态监听器

为了获得数量巨大的纷繁的数据，Senno开发了动态监听器，监听经典的Web 2.0频道（如博客，评论，YouTube频道和社交媒体）以及视频和现场直播转录。根据SennoCore发送的实时命令，可以指导动态监听器收集特定类型的信息。（见附录2）

## 问责制和反篡改机制

由于区块链技术的运用，情绪结果在传输中进行加密，然后存储在分布式数据库中并进行数字签名以确保数据完整性和重新验证，真正达到完全透明，并确保在生成情绪结果期间没有发生操作。

## SennoApps

Senno将推出一个应用市场，其中将包括多种类别的应用，如金融、医疗、营销等。这意味着终端用户将能够在他们生活的几乎各个方面都从情感分析方面获益，而开发人员则能够引入基于情感分析的APP，例如：医疗诊断应用、交易警提示、商业智能、群众智慧、营销活动效果测量等等。

## 数字身份

Senno在NEO区块链上运行，以强化数据完整性并防止操纵。通过使用NEO数字身份，Senno将能够表征可信数据通道，并使用其AI算法为其分配合适的信誉评分。

## NEO – Senno区块链

NEO智能经济是一个基于区块链的，以去中心化方式支撑智能合约、ICO和应用程序的项目。它是开源的，完整的，并且拥有专业的和敏锐的团队的支持大型社区的运转。通过使用诸如P2P联网，dBFT共识，数字证书，超导交易和交叉链式互操作等技术，NEO区块链能够以高效、安全和具有法律约束力的方式管理智能资产，是所有其他区块链所难以企及的。

### 为什么选择Neo

比特币和以太坊已经处理了一系列安全问题，但各自的区块链平台也出现了多种分歧。这就导致了各大平台需要寻求解决方案来抵御垃圾邮件和DDoS攻击，并简化其区块链。

工作证明算法需要消耗大量的能量，这就限制了初创公司可以使用以太坊的智能合约的可能性。NEO使用委托的拜占庭容错（dBFT）系统，以更好地适应智能合约。虽然它的可用性受到限制，但它确保了交易的实现，并可以支持更多的事务。

交易实现意味着在交易确认后，它将被永久记录下来，并且不能被撤销。我们相信，在财务应用方面，这种诚信远比可用性更重要。dBFT机制已被证明是可靠的，因为它在运行的时间内没有出现过一过崩溃。<sup>3</sup>

### 此外：

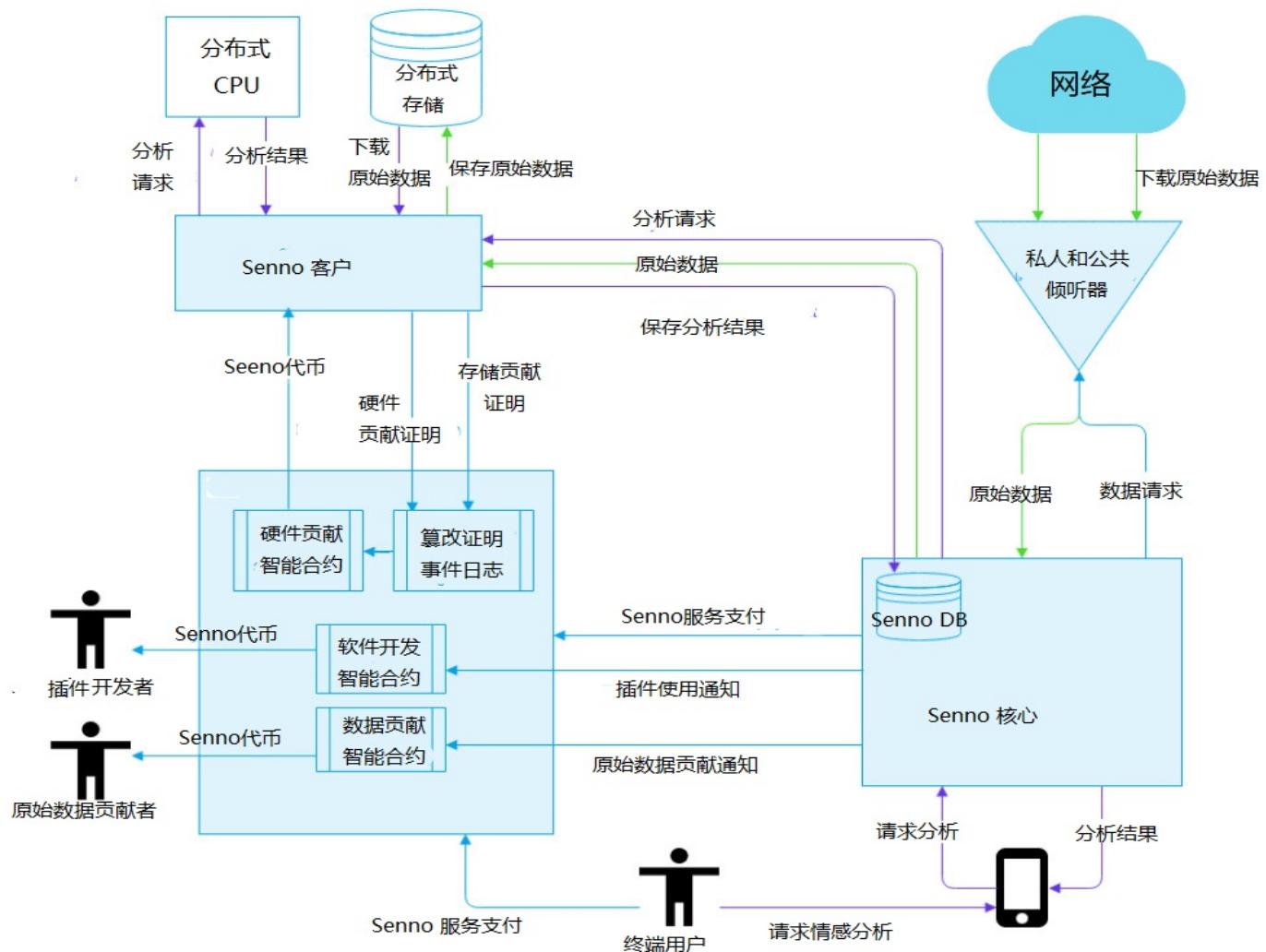
NEO的智能合约和Apps可以用C# 和Java编写和编译。未来，开发人员还可以使用Python和Go编写智能合约。这将为全球所有开发者进入该领域减少了壁垒。

NEO是量子计算机证明。业内认为，量子计算机有能力进入并破解区块链所基于的加密算术。而NEO则表明他们已经开发了称为NeoQS的反量子密码机制。

就今天而言，与以太坊每秒15笔交易相比，NEO支持高达每秒钟10,000笔交易，与以太坊相比，这是非常节能的。

# Senno基础设施概述

Senno旨在有效地推广存储在去中心化环境中的数据，并使用区块链智能合约来监督信息的完整性。Senno Network的核心是SennoCore，它充当非区块链应用的外链世界与在NEO区块链上执行的链上智能合约之间的桥梁。



Senno 网络架构



## Senno核心

- 拥有Senno网络的商业逻辑
- 处理与外部实体（SennoApps，API和SDK）的通信
- 通过区块链验证余额和对服务进行支付
- 区分现有情绪/新情绪计算的查询
- 使用AI源优先级管理数据监听器。原始数据被分成小块并保存到分布式数据存储网络中
- 与SennoClient网络通信以管理分布式硬件资源

## 数据监听器

- 充当可插式服务架构，允许添加，更改和更新数据提供者
- 允许从公共和私人渠道检索数据
- 根据Senno Core的动态优先级从指定的通道收集原始数据

## Senno客户端

- 从Senno Core中检索处理和存储任务，并在其分配的资源上执行
- 验证区块链上分布式硬件智能合约的贡献
- 构建情感分析结果并将其发送到核心位置

## NEO区块链

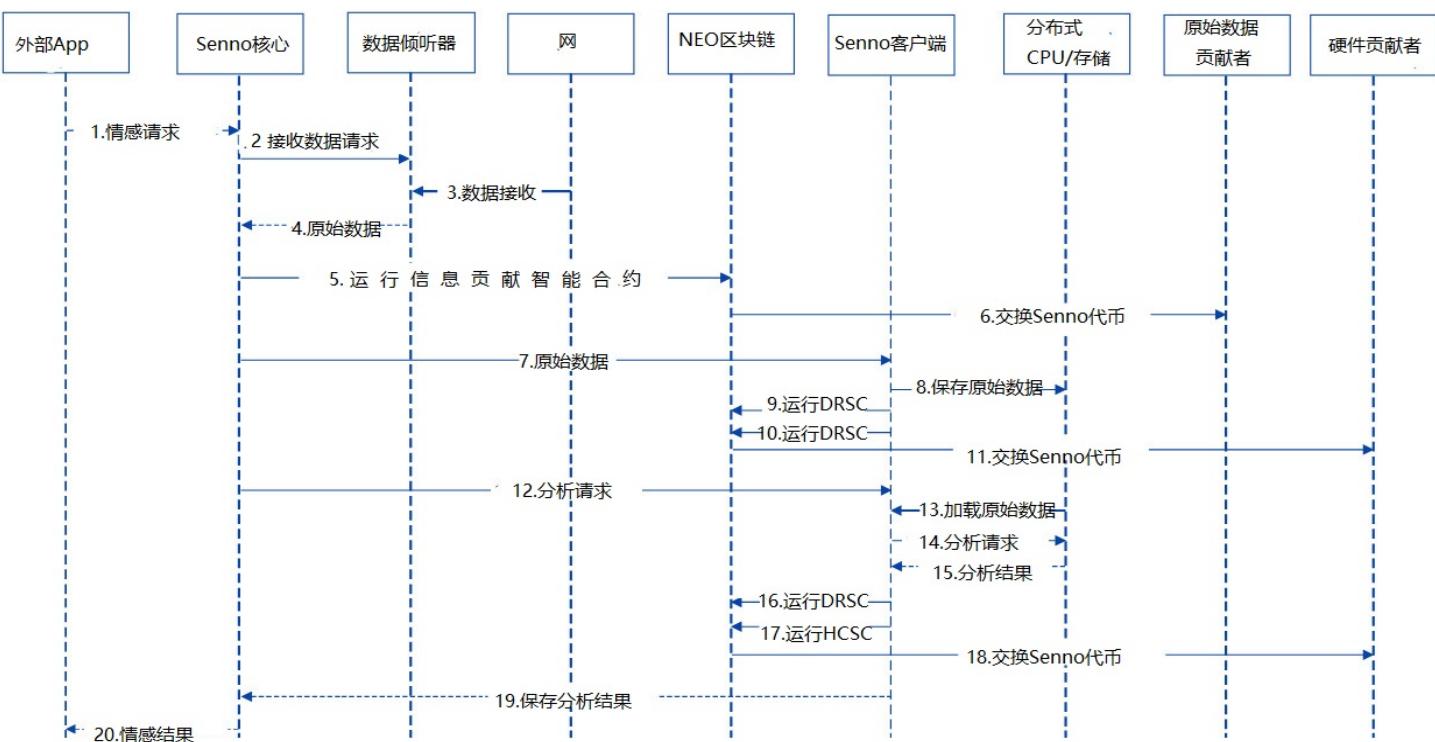
Senno的链上实施将通过以下智能合约（SC）完成：

- 硬件贡献者智能合约（HCSC）。本智能合约将完成一项交易，并在存储贡献或词法分析验证完成后向硬件提供者支付报酬。
- 数据注册智能合同（DRSC）验证硬件贡献并呼叫HCSC。
- 软件开发人员智能合同（SDSC）。此智能合约将完成一项交易，并在使用数据馈送插件后向软件开发者支付报酬。
- 信息贡献智能合约（ICSC）。本智能合约将向信息贡献者支付报酬，让Senno听众访问他们的私人频道/群组，同时确保数据不被泄露。

## 加密

- 与外部来源和第三方的通信将使用SSL / TLS进行加密
- 发送给客户端的原始数据和分析请求将使用ECC非对称加密进行加密
- 分布式分析的数据将分批进行处理，并在不同客户之间混合使用，以避免任何追踪源头的可能性
- 静态数据将使用AES算法进行加密

# 主要流程图



Senno的主要流程图

## 流程描述：

1. 第三方使用Senno的SDK或API向SennoCore发送情绪请求
2. 检查过去是否产生了情感请求（作为先前查询的结果），如果结果未产生（在SennoDB中不存在），向侦听器发送请求以检索特定的新的原始数据
3. 侦听器根据从SennoCore接收到的动态参数来检索新的原始数据并对其进行加密
4. 听众随后将新的加密的原始数据发送回SennoCore
5. 如果数据是从私人渠道收到的，信息贡献者应该通过信息贡献智能合约（ICSC）从SENNO获得代币奖励
6. SENNO代币转移给信息提供者
7. 将加密的原始数据发送给Senno客户端进行处理
8. 数据保存在分布式存储上
9. 调用数据注册表智能合约（DRSC）来验证存储的数据
10. 一旦DRSC确认存储的数据，HCSC将向存储贡献者付款
11. SENNO代币转移给硬件贡献者
12. 情绪请求从SennoCore发送到Senno客户端网络
13. Senno客户端将来自分布式存储器的原始数据加载到RAM中
14. Senno客户端将原始数据和分析请求发送到分布式CPU
15. 分析结果在分布式CPU中进行计算，数据的情感被加密并返回给Senno客户端
16. 调用数据注册智能合同（DRSC）来验证词法分析的计算是否正确
17. 一旦DRSC确认计算结果，HCSC将向存储贡献者付款
18. SENNO代币转移给硬件贡献者
19. Senno客户端将加密的分析结果发送给将其存储在SennoDB中的SennoCore
20. SennoCore以情绪结果向第三方进行反馈



## Senno背后的数学

### 词汇分类

当一个领域缺乏任何标注的数据时，仅用背景知识建立情感分类模型依然是可能的。例如，某些词的极性的词典就是一个例子。这将分配正面或负面的条款，然后可以根据每个文本的出现频率来衡量。

然后，你可以用  $P(+|D) = \frac{a}{a+b}$  计算一个元素 D 属于一个正类的概率。其中 a 是正项的出现次数，b 是负项的数量。如果  $P(+|D) > t$ ，那么文档将被视为属于正类，其中 t 表示分类的阈值。如果  $P(+|D) < t$ ，则该文件将被视为否定。

如果没有关于样本中术语的积极性和消极性的其他信息，我们假设  $t = 0.5$ 。这意味着具有更积极条款的文件将被视为正面（因为现在有先前的信息）。

我们使用这种方法作为我们的基线之一，并将其称为词法分类器。Senno 使用的词典是由 IBM 印度研究实验室为其他文本挖掘目的而开发的。它包括研究人员标记的 2,718 个正项，570 个中性项和 4,911 个负项。这个词汇不是针对任何特定领域开发的，这意味着应该使用培训示例来理解特定领域的内涵。



## 寻找贡献者

确保一个贡献者样本代表人群情绪，包括测量个人贡献者和整个样本集与人群中盛行的多数意见的一致程度。这是通过使用衡量贡献者对人群协议的估计质量的模型来实现。

$L = \{l_1, l_2, \dots, l_m\}$ ,  $l_k : x \rightarrow \{-1, 1\}$  表示贡献者集合， $\{x_1, x_2, \dots, x_t, \dots, x_n\}$  表示例子的序列。我们将  $V_{it} = l_i(x_t)$  定义为  $l_i$ 's 对  $x_t$  的投票，将  $S_t \subset \{1, \dots, M\}$  定义为贡献者  $x_t$  贡献的数量。对于每个贡献者  $l_i$ ，我们将  $c_{it}$  定义为迄今为止我们已经观察到来自  $l_i$  的贡献的次数，以及这些贡献中有多少与该贡献者一致：

$$c_{it} := \sum_{\tilde{t}=1}^t \mathbb{1}_{[i \in S_{\tilde{t}}]}, \quad a_{it} := \sum_{\tilde{t}=1}^t \mathbb{1}_{[i \in S_{\tilde{t}}, V_{i\tilde{t}} = V_{S_{\tilde{t}}\tilde{t}}]}$$

其中  $V_{S_t t} = \text{sign}(\sum_{l \in S_t} V_{lt} Q_{lt})$  是  $S_t$  贡献者  $l_i$  的质量估计中贡献者的多数票，然后定义为

$$Q_{it} = \frac{a_{it} + K}{c_{it} + 2K}$$

在这种情况下， $t$  是已经收集和标记的例子的数量，而  $K$  是产生贝叶斯收缩估计的平滑参数。这是贡献者  $l$  同意其他人群的可能性。当没有足够的数据进行更准确的估算时，数值会向下拉至  $1/2$ 。这意味着那些看到较少例子的贡献者并没有被赋予更多的权重（这些贡献者被认为能够更加确定地进行判断）。



图1 - Crowd Sense算法的伪代码 ( 附录 #1中的C#实现 )

1. **Input:** Examples  $\{x_1, x_2, \dots, x_N\}$ , contributors  $\{l_1, l_2, \dots, l_M\}$ , confidence threshold  $\varepsilon$ , smoothing parameter  $K$ .
2. **Define:**  $L_Q = \{l^{(1)}, \dots, l^{(M)}\}$ , contributor id's in descending order of their quality estimates.
3. **Initialize:**  $a_{i1} \leftarrow 0$ ,  $c_{i1} \leftarrow 0$  for  $i = 1, \dots, M$ .
4. **Loop for**  $t = 1, \dots, N$ 
  - (a) Compute quality estimates  $Q_{it} = \frac{a_{it} + K}{c_{it} + 2K}$ ,  $i = 1, \dots, M$ . Update  $L_Q$ .
  - (b)  $S_t = \{l^{(1)}, l^{(2)}, k\}$ , where  $k$  is randomly sampled from the set  $\{l^{(3)}, \dots, l^{(M)}\}$ .
  - (c) **Loop for**  $j = 3 \dots M$ ,  $j \neq k$ 
    - i.  $\text{Score}(S_t) = \sum_{i \in S_t} V_{it} Q_{it}$ ,  $l_{\text{candidate}} = l^{(j)}$ .
    - ii. If  $\frac{|\text{Score}(S_t)| - Q_{l_{\text{candidate}}, t}}{|S_t| + 1} < \varepsilon$ , then  $S_t \leftarrow S_t \cup l_{\text{candidate}}$ . Otherwise exit loop to stop adding new contributors to  $S_t$ .
  - (d) Get the weighted majority vote of the contributors  $V_{S_t t} = \text{sign}(\sum_{i \in S_t} V_{it} Q_{it})$
  - (e)  $\forall i \in S_t$  where  $V_{it} = V_{S_t t}$ ,  $a_{it} \leftarrow a_{it} + 1$
  - (f)  $\forall i \in S_t$ ,  $c_{it} \leftarrow c_{it} + 1$
5. **End**

在标注新数据的在线迭代开始时，贡献者池由3个贡献者进行初始化；我们选择2个具有最高质量估计值Q的贡献者，而其中有一个是随机选择的。这些最初的贡献者允许算法在整个贡献者质量的探索和质量估计的开发之间取得平衡。

每个贡献者必须对该示例进行投票，并为每个标签给定固定价格。来自贡献者的投票产生一个置信度分数，计算公式如下：

$$\text{Score}(S_t) = \sum_{i \in S_t} V_{it} Q_{it}$$

来自贡献者的加权多数票。一旦我们有了这个，我们需要确定Score ( St ) 是否是大多数人群投票的准确表示。如果结果不明确，我们会不断要求其他贡献者进行投票，直到我们确信贡献的结果。我们通过查看| Score ( St ) |的值来衡量我们的确定性，然后选择St候选人之外的某个贡献者具有最高质量的Qit估计值。我们使用这个贡献者的投票，看看它是否会改变加权多数投票，如果它已经包括在内，或者它会使分数 ( St ) 接近于零。

如果得分 ( St ) 接近零，那么投票几乎是平行的，这将把我们带入不确定的领域。是否添加候选贡献者的决定是根据以下公式进行的：

$$\frac{|\text{Score}(S_t)| - Q_{l_{\text{candidate}}, t}}{|S_t| + 1} < \varepsilon$$



在这个公式中， $\epsilon$ 是不确定性的准许水平， $0 < \epsilon \leq 1$ 。当(1)为真时，候选贡献者被加入成为 $S_t$ 的一部分，这使得我们贡献者对 $x_t$ 的投票。我们使用与上述相同的步骤重新计算得分( $S_t$ )，无论哪个贡献者在候选人之外具有下一个最高质量的 $Q_{it}$ 估计值。如果候选人没有被添加到 $S_t$ ，那么在这个例子中，加权多数票被指定为预测贡献。然后，我们继续分析集合中的下一个示例。

## Senno输出和提供的信息

Senno Core可以使用上述算法生成两个高度准确的变量。它还可以推荐分析考虑变量：  
(1) Buzz：- 关于该主题发生了多少讨论。  
 $\Delta B$  - 这表示新提及的总数与特定时间段的平均值之间的标准差变化量。

(2) 情绪 - 给定主题的正面或负面情绪价值。  
 $\Delta M$  - 表示计算的情绪与特定时间段的平均值之间的标准差变化量。

### 分析建议

在创建准确的实时分析建议时，我们同时考虑 $\Delta B$ 和 $\Delta M$ ，确保样本之间的间隔最短。我们的算法将测量Buzz和每分钟情绪的变化。

例如，我们于2017年3月31日对“比特币”这一主题进行了分析，当时日本金融服务局(FSA)在中午12点发布公告，称比特币很快将成为日本的官方货币。我们期望看到关于比特币的巨大利好消息，所以我们在12:15 pm看了Senno的输出。我们测量了BTC Mood在过去15分钟内的变化，并且与过去24小时内的平均15分钟间隔进行对比。我们发现了以下数据：

当前Buzz(B)代表最近15分钟=1M提及  
Avg B(以24小时为单位测量5分钟)=180K提及  
Avg $\Delta B$ (24小时)=20K评论  
 $\Delta B(B-(\text{平均}B+\text{平均}\Delta B))=800K$ 评论

BTC代币在最后15分钟 ( 12 : 00-12 : 15 ) 内从最大200K avg ( 180K + 20K std ) 参考值上升到5k buzz ( B ) 变化。在1M参考文献中，700K是积极情绪 ( P )，200K是消极情绪 ( N )，100K是中性情绪。因此信号将是：

$$Buzz(B) = \frac{\Delta B}{B} = 4 \Rightarrow \left( \frac{(1000000 - 200000)}{200000} \right) \times 100\% = 400\% \text{ UP}$$

$$Mood(M) = \frac{(P - N)}{N} = 2.5 \times 100\% = 250\% \text{ UP}$$

在15分钟的时间内都显示强正向上参考趋势。BTC价格在这15分钟内的行为与结果完全相符。

## Bitcoin (USD) Price

Closing Price

OHLC



BTC price on Mar 31, 2017 to Apr 1, 2017<sup>6</sup>

# Senno代币 ( SENNO )

Senno生态系统基于一个名为SENNO的开源加密代币。SENNO是驱动Senno语义分析平台的底层加密货币，是一个可以转换和互换的NEP-5代币，这包括允许用户获得SENNO并使用它对生态系统内服务进行支付的功能。他们可以通过积极工作（例如插件开发）或被动工作（例如从私人渠道分享数据）来获得代币。

## 用途和目的

Senno生态系统中的活动主要使用SENNO代币执行，使得代币成为Senno平台的一个组成部分，也是其经济驱动因素，其用途包括：

- 奖励硬件贡献者
- 作为软件开发商和网络资源提供商的报酬
- 向提供已请求的情绪定制插件的开发者付款
- 附属机构发送推荐后的佣金
- 订阅Senno的群众智慧数据和API使用费用

## 存款和提现

Senno将接受比特币，Neo，Altcoin，信用卡和电汇的捐款。

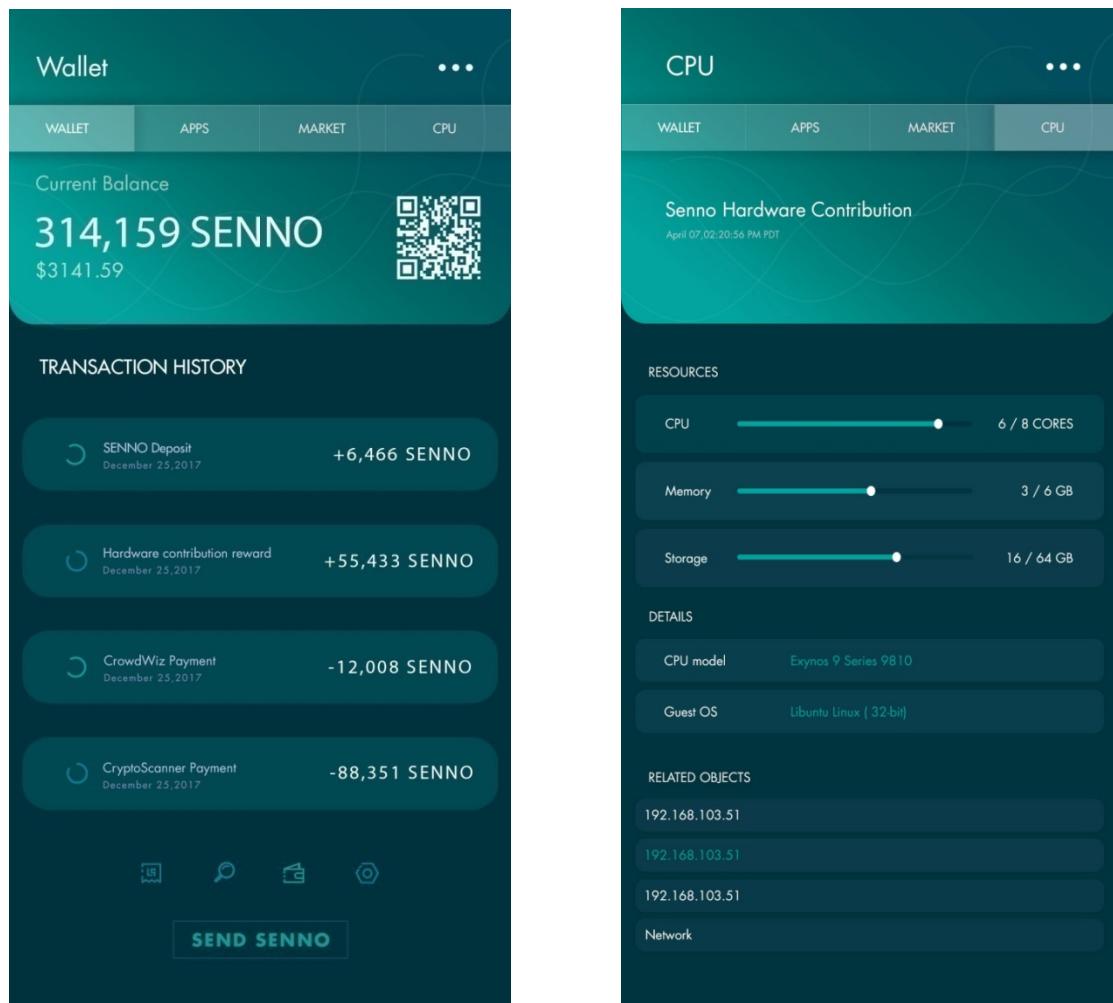
Senno将申请在所有主要的加密货币交易所进行交易。

## The SENNO 代币合约

## Senno客户端

Senno将推出可以访问其主要功能的网络和移动应用程序：

- 持有和转让基于NEP-5的SENNO代币的钱包。
- 管理向Senno网络贡献的硬件资源。
- 应用程序市场 ( SennoApps ) 将能够搜索和下载基于Senno SDK的第三方应用程序。



客户端

# 认可贡献

Senno网络旨在解决财务内容市场扩张、货币化和规模化中的主要障碍。我们创建了内置的贡献模块，奖励那些为经济提供有意义和可衡量贡献的人。这是基于我们的信念，即平台的扩展依赖于去中心化基础以及对社区和开发人员持续的激励。

## 硬件贡献奖励

Senno生态系统的根本依赖于通过在社区机器上运行的Senno客户端贡献的处理能力。30%的SENNO代币供应将被智能合约锁定并用于贡献奖励。贡献奖励适用于以下方法：

### • 信息贡献：

Senno用户可以访问他们的私人数据通道（IM APP群组电报，微信，QQ，私人新闻频道），并根据他们提供的原始数据的数量和质量，通过SENNO代币获得奖励。信息捐款将由信息贡献智能合同（ICSC）支付。

### • CPU贡献：

对于贡献的每个用来计算情绪的H / s，基于在给定时间范围内在整个网络中贡献的H / S的总量，贡献者将被授予SENNO的相对量。

### • 存储贡献：

对于每个贡献用于存储原始数据的知识库，一旦数据已经通过数据注册智能合同（DRSC）验证，贡献者将根据其在整个网络中贡献的数据库总数中的相对部分被授予SENNO代币。

硬件贡献由DRCS验证Onchain并由硬件贡献智能合约（HCSC）支付，而分布式处理和存储则作为Offchain。

### • 开发贡献：

社区成员将能够通过开发网络的功能和插件来获得SENNO代币。所开发的模块将由Senno社区Senno治理插件审批系统进行选择进入系统。根据所开发功能的复杂性和使用情况，所投票选的开发人员将向开发人员发放SENNO。



## 可持续经济

加密货币需要推动可持续经济，才能长期获得成功。Senno生态系统的价值将随着更多用户和提供商所采用的Senno在网络上的活动增加而提高。

实时情绪分析将依靠用户获取SENNO代币，需求随着采用率的增长而增长，并与网络上硬件资源提供商的数量成正比。Senno模型为向网络贡献资源的人员、插件创建者和外部开发人员提供了一种可以为SENNO代币贡献给网络的方式。

## 平台的扩展

Senno被设计为一个多层平台，用于将业务逻辑模块（执行语义分析的Senno Core）与用于从Web上检索原始数据的扩展和插件分开。通过允许任何成员针对他们认为相关的特定数据源生成量身定制的扩展，该体系结构可以使网络的覆盖范围更快，扩展更容易。贡献出新投票进入网络的会员将获得SENNO代币奖励。

## 衍生和定制版本

我们鼓励任何组织或个人更改官方参考实施，以便他们可以为Senno网络发布自己的定制化和品牌的客户端。开源定制SDK将与文档和教程一起发布，以简化开发流程。这将鼓励其他情绪分析公司使用Senno网络。

Senno的服务可用于各种应用程序，包括：

- 监测并定期报告品牌情绪
- 反映射品牌环境
- 反映目标受众
- 研究趋势
- 研究社交数据
- 自定义细分研究



此外，还将有一系列定制选项的文档，包括品牌、颜色和图形；用户界面的改变；修改相关细分的用户体验（针对具体行业的体验）；改变分析发现过程的显示指示器的排序顺序和过滤器；设置供应商联合会费用结算地址等。

这些定制和衍生的实施不会成为平台或代币的分支，而只是提供不同产品或用户体验的不同客户端。所有客户都将使用相同的网络，具有相同的特征，以及由SENNO代币支持的情绪分析。

## 加盟和合作关系

Senno对那些为网络发展做出贡献的人进行奖励，贡献者越大，获得的薪酬越高。意见领袖可能有对于在Senno在其社区中推广Senno，或者授予听众访问他们的私人渠道并获得推荐奖励感兴趣。营销教育学院可以为每个加入平台并订阅市场信息的新成员接收SENNO代币付款，并鼓励开发人员、硬件提供商和终端用户向Senno的核心添加模块，或者传播关于Senno的信息以获得额外的SENNO代币。

# 金融情绪

## 商业模式和货币化

Senno以不同的方式与世界齐头并进。首先，它会扫描社交网络、新闻网站、区块链、消息应用程序和其他数据源以获取信息，然后根据预定义的规则提取相关数据。最后，它将产生包含细化信息和趋势的情感数据流。

Senno通过汇总和分析数百万数字源实时数据，引进下一级商业智能，然后使用机器学习对企业受到的内外威胁及机会进行监控和了解。

预计到2019年，全球零售市场的销售额将超过28万亿美元，因此数百万潜在客户对安全的和高质量的商业智能数据产生需求。Senno可以以较低的成本和快速的处理时间提供这些数据。

Senno与竞争对手的比较如下：

	Industry	Open source	Decentralized	SDK presence and API	Extendibility	Price	Technology
<b>Senno</b>	Open Platform for any industry	Yes	Yes	Yes	Yes	\$240-\$6000 / Yearly	Neo
Bottlenose	BI	No	No	No	No	\$200K-\$1M / Yearly	Non-blockchain
Semantria	BI	No	No	Yes	Yes	\$12K-\$24K / Yearly	Non-BlockChain
Sether	Online advertising	Yes	No	Yes	Yes	\$ 400-\$5K / Yearly	Etherium
Saniment	Trading Signals	Yes	No	No	No	?	Etherium

Senno网络将产生两种不同的收入流，第一个种是作为B2B平台的情感数据流，第二种是作为B2C应用程序制造商：

(1) 情绪数据流 - 第三方应用程序将能够在不同的数据源上订阅和执行分析请求，包括私人（用户贡献）和公共来源。根据要求的来源和数量不同，价格也有所差异，最低为每月100美元。得益于去中心化的资源和个人贡献技术，Senno的定价明显低于传统的情绪分析服务。虽然像Semantria这样的公司每年收取12,000到24,000美元的费用，但情绪分析商业智能服务（如BottleNose）每年的成本可能高达数十万美元。相较之下，由于创新技术的应用，Senno的全面解决方案的价格范围为每年1200美元到6000美元不等。



( 2 ) 群众智慧数据应用 - Senno将与第三方合作开发基于其情感分析引擎的群众智慧应用。CryptoScanner是具有大众市场潜力的应用程序，对于加密货币交易商而言都意义重大。其订阅费每月从20美元到50美元不等，目前，其目标市场估计有将近1000万交易者。

## 将BlockChain和cryptocurrency带给大众

Senno网络的一个主要目标是向主流观众介绍加密货币分析和交易。在这个阶段，加密货币和区块链技术在技术粉丝中大受欢迎。当然，目前还有许多人没有意识到加密货币的潜力，因此，要将那些在传统市场上交易的人转移到使用基于加密货币的资产上来，是一个巨大的挑战。

加密数字代币由可通过互联网访问的区块链网络支持，因此，新时期下市场将主要在线上领域逐渐扩散。传统的市场交易者和金融领域的新手务必学习如何处理大量的信息，因为新的财务分析工具，广泛的数据源以及有前景的加密货币，每天都在不断诞生。

Senno的创建就是为了分析这一切，并利用人群的智慧帮助其社区找出他们应该对哪些加密货币进行交易，以及哪些代币能使他们赚钱。Senno强大的算法能够产生即时的情绪分析和趋势，而去中心化的点对点计算机网络能力使其能够达到传统金融市场从未见过的准确率。

## CryptoScanner - Senno的第一个应用程序

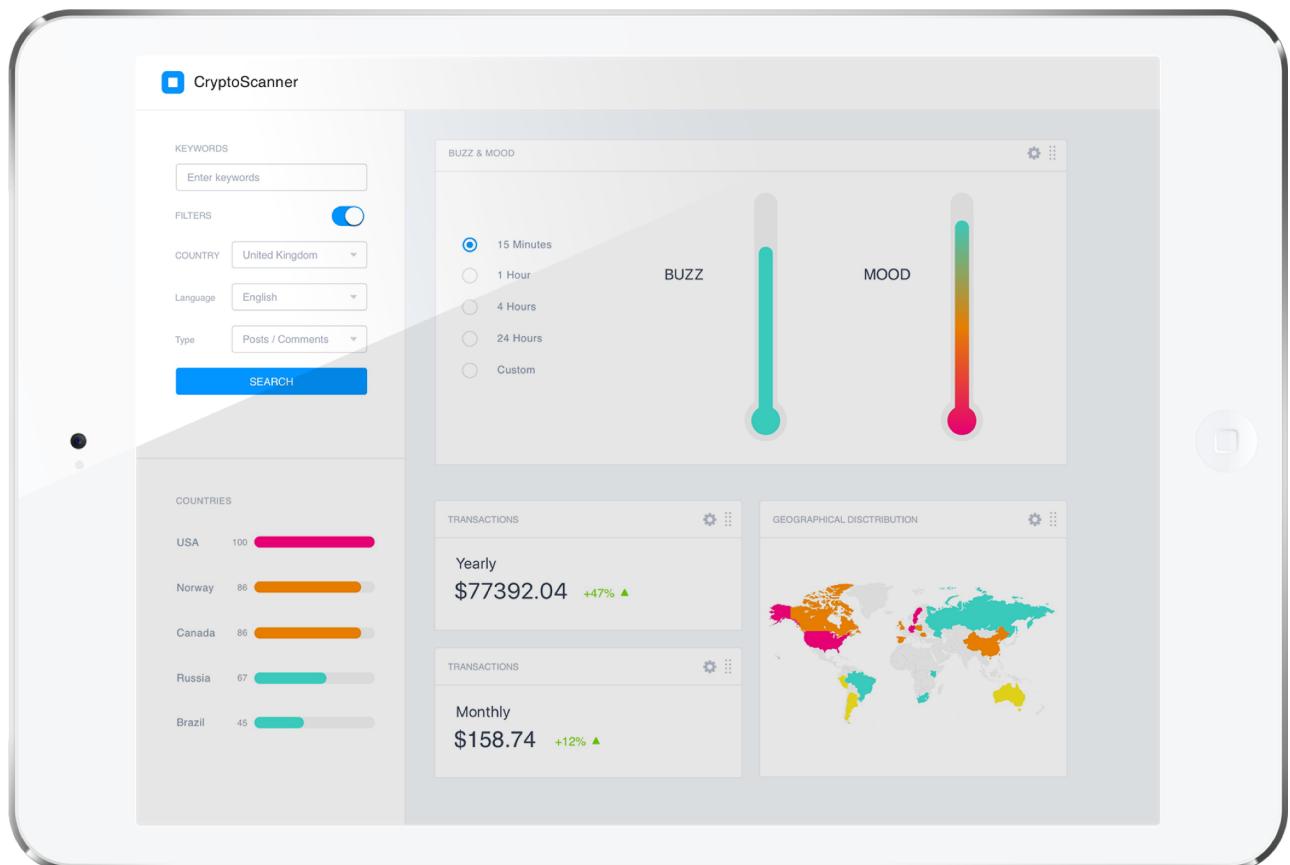
从金融市场诞生的那一刻起，投资者们就从未停止征服。多年来，人们已经进行了一系列技术来尝试，并屡获突破，包括分析数据和寻找模式，或者对公司财务和经营进行深入研究，以准确评估其价值。

情感分析则是另一个关键策略，它将深入群众的大脑去窥探他们对资产的看法。预测动向是更好地为交易进行服务，这种方式在理论上很理想，但需要对人们的感受进行准确的测量和预测。直到现在，人们对这些技术的投资依然持乐观态度。

## CryptoScanner的功能

Senno将与第三方合作，为其平台开发客户端并进行发布。这是一个功能齐全，能够用加密数字货币进行实时交易的应用程序，能够精确显示每一个Buzz和Mood的时间。

CryptoScanner将是开放源代码，用于展示用户端模块的实施情况，包括通过NEO对Senno智能合约进行链上访问，从去中心化云存储访问语义内容，报告和分析等。



*CryptoScanner*

## 主要特点：

- 实时扫描加密货币渠道。
- 多种加密货币的对比。
- 按地区，国家，语言和频道类型进行筛选。
- 基于情绪和Buzz变化的可配置预警。
- 领先交易所的API接口。

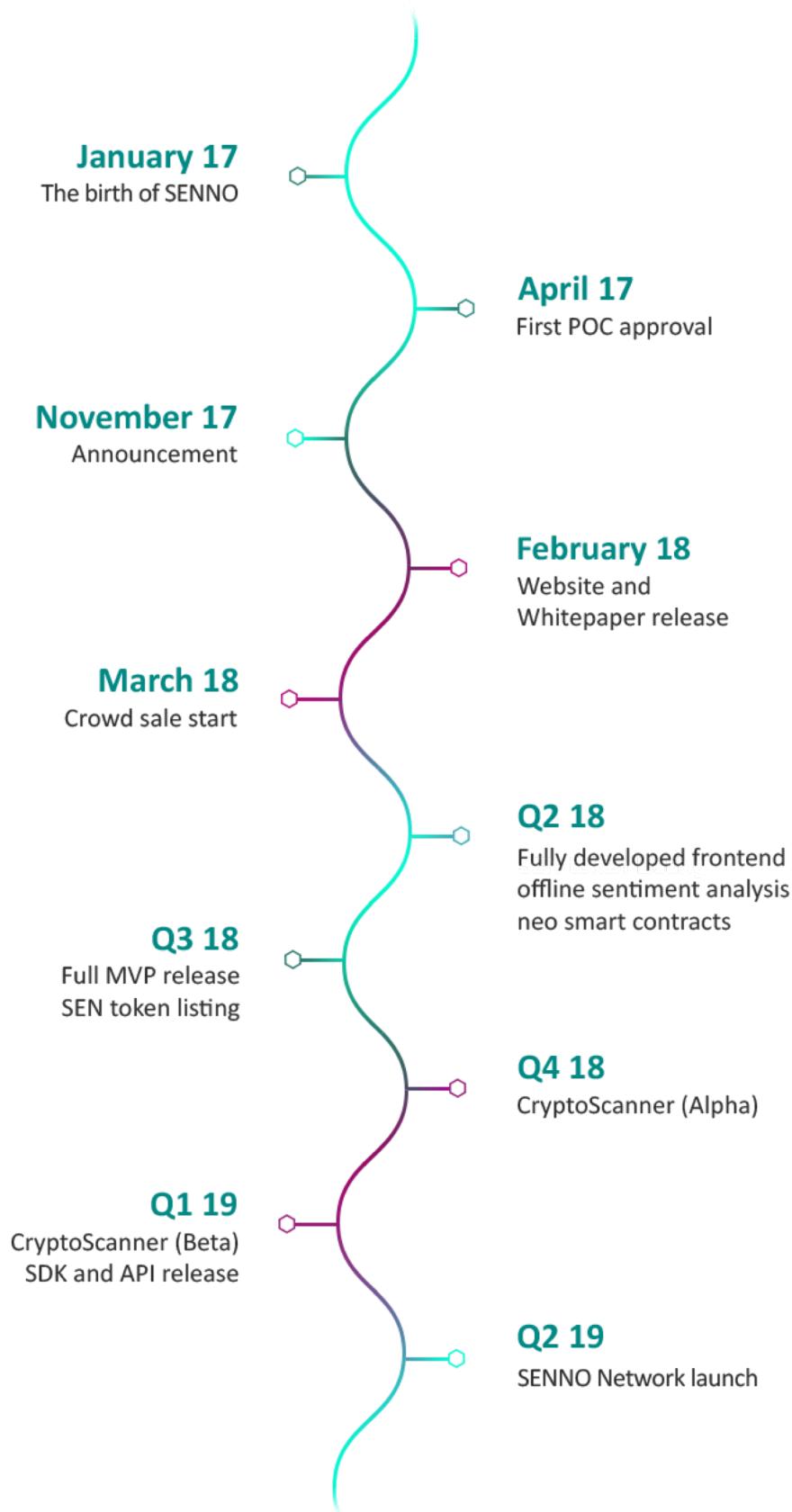
CryptoScanner可以在所有主流的操作系统上运行 - Windows，macOS，Linux，Android和iOS以及基于HTML5网页的客户端。

## 参考实现

CryptoScanner将在GitHub上作为开源软件发布，并作为Senno网络和客户端的参照。为了保持与Senno网络的兼容性，应用程序必须注意以下几个方面：

- 1) 将SENNO作为所有网络活动的首选代币。
- 2) Senno平台官方使用的NEO智能合约。
- 3) 遵从Senno P2P协议共享硬件资源。

# ROADMAP



## SENNO的代币销售活动

为了实现Senno的财务线路，Senno将出售4B SENNO代币。在销售的早期阶段，这些代币将以低价出售。在销售结束时，分销的SENNO代币将作为流动性供应。根据长期发展计划，部分供应将预先分配给Senno网络。

代币销售活动将分为三个阶段，每个阶段提供不同级别的折扣

- 私人销售
- 预售（400万美元上限）
- 人群销售（2500万美元）

参与方式：NEO，BTC，alt硬币，信用卡和电汇。

预售将于2018年3月31日开始，并将持续30天或直至达到400万美元的上限。参与说明可能会在不透的销售阶段更新，并在Senno.io网站上发布。

# 代币销售经济

- 1) 软顶 : 200万美元
- 2) 硬顶 : 2500万美元
- 3) 总量 : 100亿 SENNO
- 4) 销售阶段 :

## A. 种子期 ( 预售 )

基础价格 : \$0.01 = 1.6 SENNO

贡献	收益	收益兑现	SENNO/cent
\$500-\$15K	20%	1 Month	1.92
\$15K-\$30K	25%	1 Month	2.00
\$30K-\$90K	30%	1.5 Month	2.08
\$90K-\$250K	35%	1.5 Months	2.16
\$250K-\$500K	40%	2 Months	2.24
\$500K-\$1M	45%	2.5 Months	2.32
\$1M+	50%	3 Months	2.40

## B. 大众销售

基础价格 : \$0.01 = 1.6 SENNO

贡献	达到 \$50k	\$50 - \$100k	\$100k+	收益兑现
最初 48 小时	10%	15%	20%	1 个月
第 1 周	5%	8%	10%	0.5 个月
第 2 周	3%	5%	7%	0.5 个月
第 3 周	1%	2%	3%	-
第 4 周	0%	0%	0%	-

- 5) 剩余的未售出的代币将被摧毁。

- 6) 代币归属期 :

- 奖励归属只适用于所有奖金和奖金代币。当SENNO在交易所上市时，归属期开始
- 合作伙伴和顾问 : 24个月归属，6个月后，代币每3个月按比例发布
- 员工 - 36个月归属，1年后，代币按比例每3个月发布一次

- 7) 销售结束后30天，锁定转账。

## 资金使用

企业资金在项目中的使用情况如下



## Senno代币分配



# 我们的团队



**ELAD PELED**  
创始人

区块链专家，连锁企业家，  
LibraTrade™Inc的创始人，该公司是  
一家金融技术公司，于2008年开发出  
世界上第一个智能手机交易平台。他  
任职于IDF情报部门，掌握了多项专  
业技能。



**RUDY ZAKUTO**  
联合创始人兼首席技术官

技术和区块链专家，对区块链架构有  
深入认识，为NEO和一致性算法开发  
了智能合约。在网络营销专业领域  
拥有超过20年的开发经验。



**LENNON TAM**  
亚太区首席运营官

金融科技专家，亚太市场区块链行业  
领导者；在银行业担任财富经理超过  
7年，同时也是一位经过认证的理财  
规划师和百万圆桌行业协会会员。



**CARLOS MANTILLA GUEVARA**  
区块链开发者



**KIRILL LUKIANENKO**  
高级开发者



**ALEXANDER VOLKOV**  
研发领导者



**TATYANA ARKHIPOVA**  
高级开发者



**NOA OMER**  
平面设计



**WES ZHANG**  
客户端开发



**CLAUDIA CASTANEDA**  
公关经理



**CHARLES C FLY**  
服务端开发

## 顾问团队



**OFIR GARTNER**  
STOX-创始人  
Invest.com - 联合创始人  
DunaPortal - 联合创始人



**MARC KENIGSBERG**  
Blocksmarter - 创始人  
CoinJanitor - 创始人



**MICHAEL GREENBERG**  
Finance Magnate-创始人及CEO  
Trad.io & CrowdWiz.io -顧問



**VESEVOLOD PELIPA**  
SolarLab - CEO  
Alvion Europe - 前首席技术官



**ALEXANDER CHALY**  
技術顧問  
Shakuro -前首席技术官  
SalesTech - 技术總監



**VAN YUEN**  
財務顧問  
中国银行 – 企業客户经理



**GARY BERNSTEIN**  
CoTrader -创始人及CEO  
RefChain –前创始人及CEO

## 合作伙伴





## 风险因素

拥有和使用代币是对平台成功的信任，也是早期的一种支持手段。然而，所有代币销售中的所有参与者都必须知道，参与行为具有一定的风险性。

以下是与Senno业务和代币销售活动有关的风险因素：

- Senno是一个复杂的软件平台，如果由于不可预见的发展障碍出现，它的推出可能会大大延迟。
- 市场竞争可能会带来不同的情感分析解决方案，并导致Senno失去市场份额，最终无法实现其业务目标。
- 国际法律和法规可能导致SENNO贸易受到影响。
- SENNO代币的使用可能受到政府机构的审查。
- SENNO代币的所有权可能受新的和不可预测的税法影响，从而使SENNO的利益受到侵害。
- Senno可能无法成功地为SENNO代币创造必要的接受度，可能导致流动性较弱。
- 随着项目的进展，本白皮书中概述的立场和计划可能会改变。

## 监管策略

作为一个情感分析平台，Senno设计的去中心化基础设施适用于各个行业和市场的应用。因此，进行大量活动可能会受到政府机构的监管审查。这种庞大而多样的组织需要遵守证券交易、金融机构和货币服务业务以及数据版权和用户隐私的规定。遵守这些规定中的每一条，有时在每个辖区内都需要经验并且巨大开销。

在去中心应用中，没有一个实体能够控制其运营所需的基础设施。这就允许将应用内容分离出基础设施。由此产生的结构具有模块化隔离的优点：系统中的单独功能可以彼此独立运行，并可能由不同的供应商提供。

在遵守法规方面，模块化隔离为企业和个人带来了优势：每个企业只需遵守特定领域的法规，降低创建所需专业知识的成本，新业务也更容易启动。

## 参考文献

- [1] S. eyda Ertekin, Haym Hirsh, Cynthia Rudin MIT Sloan School of Management, Massachusetts Institute of Technology Department of Computer Science, Rutgers University, *Approximating the Wisdom of the Crowd*, 2015, pp.2-6  
[http://web.mit.edu/seyda/www/Papers/NIPS11Workshop\\_ApproximatingWC.pdf](http://web.mit.edu/seyda/www/Papers/NIPS11Workshop_ApproximatingWC.pdf)
- [2] A. J. Quinn and B. B. Bederson, "Human computation: a survey and taxonomy of a growing field," in *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI)*, 2011, pp.1403–1412.
- [3] Bo Pang and Lillian Lee. "Opinion mining and sentiment analysis" *Foundations and Trends in Information Retrieval* Vol. 2, No 1-2 (2008) 1–135  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.147.2755&rep=rep1&type=pdf>
- [4] G. Ramakrishnan, A. Jadhav, A. Joshi, S. Chakrabarti, and P. Bhattacharyya. "Question answering via Bayesian inference on lexical relations. In *ACL Workshop on Multilingual Summarization and Question Answering*", 2003 <https://aclanthology.info/pdf/W/W03/W03-1201.pdf>
- [5] Bing Liu. "Sentiment Analysis and Opinion Mining", Morgan & Claypool Publishers, May 2012.  
<https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf>
- [6] Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, Bing Liu "Sentiment Analysis in Social Networks" 2017

## 附录 #1 Crowd Sense implementation

```

using System;
using System.Collections.Generic;
using System.Linq;
namespace Main
{
    class CrowdSense
    {
        public class Example { }
        public class Labeler
        {
            public double[] QualityEstimate { get; set; }
            public double[] NumberofContributions { get; set; }
            public int[] NumberOfContributions { get; set; }
            public int[] Votes { get; set; }
        }
        public void Calculate(List<Example> examples, List<Labeler> labelers, double threshold, double smoothingParameter)
        public void Calculate(List<Example> examples, List<Labeler> labelers, double threshold, double smoothingParameter)
        {
            Random rnd = new Random();
            Random randomLabelerIndex = new Random();
            var m = examples.Count;
            var n = examples.Count;
            Init(labelers, n);
            for (var exampleIndex = 0; exampleIndex < n; exampleIndex++)
            Init(labelers, n);
            for (var exampleIndex = 0; exampleIndex < n; exampleIndex++)
            {
                CalculateQualityEstimate(examples[exampleIndex], smoothingParameter, exampleIndex);
            }
            CalculateQualityEstimate(labelers[0], labelers[1], labelers[2], ..., labelers[n - 1], exampleIndex, rnd, m);
            UpdateNumberedDablersbutConsistentdablers(exampleIndex, rnd, m);
        }
        UpdateNumberofContributions(selectedLabelers, exampleIndex);
    }
    public static void Init(List<Labeler> labelers, int n)
    {
        foreach (var labeler in labelers)
        {
            labeler.QualityEstimate = new double[n];
            labeler.NumberofContributions = new int[n];
            labeler.NumberofContributions = new int[n];
            labeler.QualityEstimate = new double[n];
            labeler.NumberofContributions = new int[n];
        }
        labeler.NumberofContributions = new int[n];
    }
    public static void CalculateQualityEstimate(List<Labeler> labelers, double smoothingParameter, int i)
    {
        foreach (var labeler in labelers)
        labeler.QualityEstimate[i] = (labeler.NumberofContributions[i] + smoothingParameter) /
        foreach (var labeler in labelers)(labeler.NumberofContributions[i] + 2 * smoothingParameter);
    }
    labeler.QualityEstimate[i] = (labeler.NumberofContributions[i] + smoothingParameter) /
    public static List<Labeler> GetSelectedLabelers(double threshold, Random rnd, int m)
    {
    }
    labelers.Sort((x, y) => x.QualityEstimate[i].CompareTo(y.QualityEstimate[i]));
    public static List<Labeler> GetSelectedLabelers(List<Labeler> labelers, double threshold, int i, Random rnd, int m)
    {
        var randomLabelerIndex = rnd.Next(2, m);
        var selectedLabelers = new List<Labeler> { labelers[0], labelers[1], labelers[randomLabelerIndex] };
        for (var j = 2; j < m; j++)
        labelers.Sort((x, y) => x.QualityEstimate[i].CompareTo(y.QualityEstimate[i]));
        var randomLabelerIndex = rnd.Next(2, m);
        var selectedLabelers = new List<Labeler> { labelers[0], labelers[1], labelers[randomLabelerIndex] };
        for (var j = 2; j < m; j++)
        {
            var current = (Math.Abs(score) - labelers[j].QualityEstimate[i]) / (selectedLabelers.Count + 1);
            if (current < threshold) continue;
            var score = selectedLabelers.Sum(sc => sc.QualityEstimate[i] * sc.Votes[i]);
            selectedLabelers.Add(labelers[j]);
            var current = (Math.Abs(score) - labelers[j].QualityEstimate[i]) / (selectedLabelers.Count + 1);
            else
                selectedLabelers.Add(labelers[j]);
        }
        return selectedLabelers;
    }
    public static void UpdateNumberofContributions(List<Labeler> selectedLabelers, int exampleIndex)
    {
        var sign = Math.Sign(selectedLabelers.Sum(sc => sc.QualityEstimate[exampleIndex] * sc.Votes[exampleIndex]));
        foreach (var sel in selectedLabelers)
        {
            sel.NumberofContributions[exampleIndex] += (Math.Sign(sel.Votes[exampleIndex]) == sign ? 1 : 0);
        }
        foreach (var sel in selectedLabelers)
        {
            sel.NumberofContributions[exampleIndex] += (Math.Sign(sel.Votes[exampleIndex]) == sign ? 1 : 0);
            sel.NumberofContributions[exampleIndex] += 1;
            sel.NumberofContributions[exampleIndex] += (Math.Sign(sel.Votes[exampleIndex]) == sign ? 1 : 0);
        }
    }
}

```

## 附录#2 Data Listener Interfaces

```
using System;
using System.Collections.Generic;
namespace SocialNetwork
{
    public class BaseEntry
    {
        public string ID { get; set; }
        public DateTime Date { get; set; }
        public string Text { get; set; }
    }
    public interface IFacebookExtractor
    {
        List<BaseEntry> GetPageItems(string pageId, int count);
        List<BaseEntry> GetComments(string objectId, int count);
        List<BaseEntry> GetMentions(string pageId, int count);
    }
    public interface ITwitterExtractor
    {
        List<BaseEntry> GetUserTweets(int userId, int sinceId, int maxId, int count);
        List<BaseEntry> GetHashtagTweets(string hashtag, int sinceId, int maxId, int count);
        List<BaseEntry> GetMentionTweets(int userId, int sinceId, int maxId, int count);
        List<BaseEntry> GetListTweets(int listId, int sinceId, int maxId, int count);
    }
    public interface IGooglePlusExtractor
    {
        List<BaseEntry> GetUserActivities(string userId, int count);
        List<BaseEntry> GetActivityComments(string activityId, int count);
    }
    public interface IInstagramExtractor
    {
        List<BaseEntry> GetUserMediaCaption(int userId, int minId, int maxId, int count);
        List<BaseEntry> GetHashtagMediaCaption(string tagName, int minTagId, int maxTagId, int count);
        List<BaseEntry> GetMediaComments(int mediaId);
    }
    public interface ILinkedInExtractor
    {
        List<BaseEntry> GetUserShares(string ownerId, int offset, int count);
        List<BaseEntry> GetUserArticles(string ownerId, int offset, int count);
        List<BaseEntry> GetComments(string ownerId, string commentId, int offset, int count);
    }
    public interface IRedditExtractor
    {
        List<BaseEntry> SearchSubreddits(string subreddit, string query, int count);
        List<BaseEntry> GetComments(string subreddit, string link, int count);
    }
    public interface ITumblrExtractor
    {
        List<BaseEntry> GetBlogPosts(int blogId, int offset, int count);
    }
    public interface IVkExtractor
    {
        List<BaseEntry> GetUserPosts(int ownerId, int offset, int count);
        List<BaseEntry> GetComments(int ownerId, int postId, int offset, int count);
    }
    public interface ITelegramExtractor
    {
        List<BaseEntry> GetHistory(int chatId, int offset, int count);
    }
}
```

Copyright © 2017 Senno.io

**DISCLAIMER:** This SENNO.IO Technical White Paper is for information purposes only. Senno.io does not guarantee the accuracy of or the conclusions reached in this white paper, and this white paper is provided "as is". Senno.io does not make and expressly disclaims all representations and warranties, express, implied, statutory or otherwise, whatsoever, including, but not limited to: (i) warranties of merchantability, fitness for a particular purpose, suitability, usage, title or non infringement; (ii) that the contents of this white paper are free from error; and (iii) that such contents will not infringe third-party rights. Senno.io and its affiliates shall have no liability for damages of any kind arising out of the use, reference to, or reliance on this white paper or any of the content contained herein, even if advised of the possibility of such damages. In no event will Senno.io or its affiliates be liable to any person or entity for any damages, losses, liabilities, costs or expenses of any kind, whether direct or indirect, consequential, compensatory, incidental, actual, exemplary, punitive or special for the use of, reference to, or reliance on this white paper or any of the content contained herein, including, without limitation, any loss of business, revenues, profits, data, use, goodwill or other intangible losses.

All rights reserved to Senno Holdings LTD.

New Trend Centre 704/811

Prince Edward Road East

San Po Kong

Hong Kong

[info@Senno.io](mailto:info@Senno.io)