

# 目录

<b>前言</b>	<b>4</b>
工作组介绍	4
蓝皮书介绍	5
蓝皮书发布之后	5
<b>背景介绍</b>	<b>5</b>
工作组的组建过程	5
公众社区的积极参与	6
蓝皮书的撰写过程	6
<b>基本理念</b>	<b>7</b>
基本假设	7
目标受众统计	7
生态共同需求	8
不确定的可持续性	8
融合集体智慧	8
核心原则	8
开发者优先	9
前瞻性	9
支持去中心化	9
可行性与实用性	9
慎重规划	9
无障碍和低门槛	9
可互操作性	10
<b>API 现状分析</b>	<b>10</b>
原生 EOSIO API	11
Chain	11
History	11
State History	11
Trace	11
社区开发的 EOSIO API	11
API 类型	12
衍生状态API	12
流媒体API	12
交易历史记录API	12
解决方案	12
Hyperion	13
Chronicle	13

Light API	13
Firehose	13
Dfuse	13
Roborovski	14
<b>提案详情</b>	<b>14</b>
提案 1: API调研和标准制定	15
具体提案	15
需求清单	15
基本原理	16
工作职责	16
研究与评估	16
系统架构	16
规范文档	17
协调发展	17
资助方式	17
为ARS团队提供资金	17
软件开发	17
成本预估	17
相关建议	18
提案 2: 交易执行周期	19
具体提案	19
交易资源成本预估	19
改进主观计费方式	19
交易重写	20
交易最终状态	20
需要被考虑在内的影响因素	20
网络改进需求	20
项目代码库	20
项目指南	21
成本预估	21
相关建议	21
提案 3: 专用 API	21
具体提案	21
软件维护	22
支持现有运营商	22
提高可用性	22
功能开发	22
基本原理	22
用户体验风险	22
目前所处形势	23

需要被考虑在内的影响因素	23
EOSIO 更新	23
API架构的时间安排	23
缺少高质量的开发专家	23
网络增长	23
供应链问题	24
对资助对象的工作要求	24
透明度	24
代码开源	24
成本预估	24
开发	24
运营	25
相关建议	26
提案 4:中心 API 服务	27
具体提案	27
基本原理	27
建议提供的服务	27
公共产品	27
付费服务	28
资金考虑	28
交付方式分析	29
方案1:将所有责任委托给签约团队	29
方案2:网络负责领导,外部团队管理运营	30
方案3:由网络领导和运营	30
成本预估	30
品牌塑造	30
组织搭建	30
计费基础设施	31
部署方式	31
运营/支持	31
相关建议	32
确定方法	32
规划前期投资	32
账户增长	32
提案 5:分布式 API	32
具体提案	32
基本原理	33
成本预估	33
相关建议	33
提案 6:区块链数据库	33
具体提案	33

基本原理	34
成本预估	34
相关建议	36
提案 7: Rosetta	36
路线图	37
成本预估	37
相关建议	38
提案 8: The Graph	38
具体提案	38
基本原理	38
需要被考虑在内的影响因素	39
成本预估	39
相关建议	40
<b>结语</b>	<b>40</b>
<b>致谢</b>	<b>42</b>

## 前言

EOSIO 是自 2017 年 4 月以来一直在开发更新的第三代区块链技术平台。尽管生态对核心代码库的开发进行了大量投资，但 EOSIO 仍有一个关键需求从未完全得到满足，那就是许多外部应用开发者苦苦寻求的实用的 EOSIO API。由于对 EOSIO 相关数据的访问限制，开发人员往往需要花费大量精力来满足其应用程序的数据要求。

## 工作组介绍

为了改善这一状况，EOS 网络基金会首先于 2021 年 9 月启动成立了 API+ 工作组。EOS Nation、EOS Rio 和 Greymass [这三个团队](#) 凭借过往丰富的开发、操作和使用 EOSIO API 的知识及经验，被选中并且齐聚一堂，负责领导 API+ 核心工作组的工作。

EOS 网络基金会要求这三个团队创建一份关于 EOSIO API 的「蓝皮书」。蓝皮书内容将融合传统意义上的白皮书、黄皮书、提案请求和路线图等，在落实 EOS 路线图方面发挥重要作用，是一个评估 EOS 生态现状、识别潜在问题并为未来发展制定路线图的绝佳机会。关于工作组的其余工作由团队协商决定，最终目标是在 2022 年初完成并发布蓝皮书，完美完成任务。

## 蓝皮书介绍

API+工作组基于项目小组成立诉求，对EOSIO API进行了初步研究，以进一步了解 EOSIO 生态目前提供的API服务现状，以及应用程序开发人员的需求。然后，我们将研究范围扩展到其他分布式账本技术及其解决方案的设计方法，从而进一步了解目前整个行业的诉求。这份蓝皮书是研究的成果结晶。

在蓝皮书的开头部分，我们将深入介绍工作组的工作及[历史背景](#)。然后我们将介绍一些关于我们所研究领域的[基本原理](#)，用来帮助读者进一步理解我们提出优化建议背后的思考。接着我们在文档中展现了当前的[API现状](#)，让大家了解目前生态正在使用的技术名称，以及存在的每个解决方案选项。蓝皮书的最后，我们整理罗列了一份可供参考的[建议清单](#)，并给出了[结论](#)，描述EOSIO可采取的下一步的行动。

## 蓝皮书发布之后

随着API+工作组蓝皮书的发布，关于「接下来要怎么做」将成为 EOS 网络基金会和 EOSIO 生态系统的重点议题。大家需要以批判的眼光看待蓝皮书内容，并公开讨论其中提出的研究和建议。而且我们必须就每个提案的执行优先级、资金分配和负责方(团队)做出最终决定。

我们欢迎大家充分参与讨论，并提出建议和意见。

## 背景介绍

蓝皮书内容由[三个独立的团队](#)一起工作，从头开始研究并撰写。这一过程并没有提供固定的研究方向，因此也为团队提供了一个独特的学习机会。本节将概述三个团队是如何结合在一起，以及在撰写过程中，不同目标受众的公众参与过程。

## 工作组的组建过程

2021年9月，EOS网络基金会召集了History+工作组中的EOS Nation、EOS Rio和Greymass团队组建了API+工作组。由于在EOSIO API方面拥有足够的了解和历史解决方案的经验，EOS网络基金会选择了这三个团队组建API+工作组。这个小组在EOSIO和更广泛的区块链生态系统中，拥有多年软件构建、基础设施运营和部署、提出历史解决方案的经验。

撰写蓝皮书的工作，使这个新成立的工作组有机会开展自我组织和协同工作方面的尝试。团队的第一步是相互熟悉，并确定如何准确地处理需要完成的工作。通过共享各自团队的组成、管理

的 API 操作以及开发、运营的历史解决方案等方面的信息，工作组最终完成了团队之间的自我组织和相互熟悉。

起初，EOS网络基金会的想法是深入研究现有的历史解决方案，并为历史解决方案的可持续性管理推荐一条可行的道路。但在研究开始后，各工作组发现，要解决历史解决方案的一些缺陷，工作组的研究还需要拓展至与API相关的更广泛的范围，于是大家一致决定将工作组改名为API+工作组，并扩大研究范围。

由于当时离蓝皮书交付日期只剩几个月时间，团队成员决定首先确定EOSIO中最需要关注的领域。从历史解决方案开始，这场头脑风暴引发了每个API相关提案对于目前现状以及未来优化方向的思考与讨论，并最终促成了蓝皮书中[基本原理](#)部分的创建，该部分主要强调了目前而言哪些原理是最重要的。

随着研究的顺利进行，[API+工作组在2021年11月宣布成立](#)。这拉开了公众参与API+蓝皮书研讨阶段的序幕，工作组可以和EOSIO社区之间进行积极互动。

## 公众社区的积极参与

API+工作组为了收集尽可能多的信息，采取了一系列的方法。

最直接的方法是：向已经参与EOSIO生态系统的相关者征集意见。API+工作组接触了几十个不同的团队，并通过问题对话，获取了这些团队最大的愿望和相关需求的信息。这些对话让团队收获了社区丰富的反馈意见和需求建议。

API+工作组还积极听取了EOSIO公共讨论频道的所有对话，了解在EOSIO上构建的开发人员是如何使用API的。API+工作组同时参与了Telegram、WeChat、Discord和Twitter的讨论。这让API+工作组感受到了开发人员目前面临的痛苦和强烈的需求，并为改善EOS开发环境提供了机会。

工作组收集到的信息主要来源于需要使用 EOSIO API 服务的各种专业人士，包括基于 EOSIO 构建服务的应用程序、交易所和其他组织的开发人员或运营商。

## 蓝皮书的撰写过程

工作组收集到的所有研究数据和反馈，都按照不同主题和方向建立了细分的文件夹并留档，并从中筛选出社区呼声最高的、确定的主要研究议题。这些议题被交给工作组成员进行进一步研究。工作组带回并审查了其他的调查结果，并讨论了潜在的解决方案，这部分内容一并纳入本文。

工作组确定了其认为可行的解决方案后，另一部分工作组成员便开始合作撰写内容。工作组中研究者和撰写者之间进行的不断交流反馈和意见修改，促成了可行提案的产生。然后，所有的提案

被合并到蓝皮书草案中进一步完善，让API+工作组的蓝皮书成为一份有凝聚力的文档，并且能够呈现关于解决方案的多种想法。

在拥有了一份包含所有提案和提案理由的文档之后，工作组还需要进行适当的删减工作，以便从繁杂的需求中筛选出真正重要且急迫的提案。许多提案重复了相同的主题，试图解释相同的概念，并阐述了可行的解决方案。因此，本文还设置了非提案内容部分，以展现这部分内容。

## 基本理念

在工作组与生态利益相关者的讨论和随后的研究中，大家对应该如何从根本上解决EOSIO API问题，提出了很多问题。本节概述了[基本假设](#)，以及我们在制定提案时力求遵守的[原则](#)。

## 基本假设

EOSIO API 必须能够满足广泛的需求。虽然创建一个能够满足所有需求的单一解决方案，几乎是不可能的，但可以采取措施确保大多数常见的需求能够得到满足。

为了确定如何尽可能广泛地满足需求，我们进行了一次需求评估。基于需求评估，我们在蓝皮书中做出了以下假设。

## 目标受众统计

本蓝皮书中概述的目标人群是，将基于 EOSIO 的区块链，集成到应用程序和服务中的任何一方。这部分目标人群包括但不限于：

- 应用程序开发者
- 数据分析师
- 交易所/托管服务提供商
- 决策人员/领导角色
- 运营商和服务提供者
- 协议/工具开发人员
- 智能合约开发者

以上类型的个人/团队构成了主要的利益相关者，当EOSIO API基础设施更改变化时，这些人也将会受到影响。

## 生态共同需求

在与各利益相关者的讨论中，许多需求几乎是所有各方共同的需求。因此工作组假设：由于每个参与者的反馈都很相似，因此这些未满足的需求在整个生态系统中经常被发现。

虽然当今现有的 API 解决方案满足了许多常见的需求，但 API 通常会丢失重要信息、或提供不相关的信息，或者根本无法被广泛访问。这给开发人员和运营商都带来了挑战，因为开发人员被迫去解决这些低效率问题，提供 API 服务的运营商也必须面对很多不必要的负担。

## 不确定的可持续性

许多API是作为公共产品运作的，面向所有访问的人免费提供数据。如果没有专门设计的商业模式来帮助这个生态系统的可持续发展，EOSIO API基础设施的关键组成部分的运作就会充满不确定性。

大部分时候，这些服务的运营商提供的服务是亏损战略的一部分，访问权限往往绑定其他更有利可图的产品和服务。这将导致API这种公共产品的可靠性，最终取决于运营商的其他服务产品盈利情况。这种情况会带来很大风险，即单个项目的成功或失败会导致区块链及其数据的可访问性出现问题。

为确保这些服务在短期内广泛可用，工作组假设 API 解决方案的许多运营商需要资助，直到运营商能够自给自足，或生态中存在足够分散的替代方案的出现。

## 融合集体智慧

由于基于EOSIO的开发只占区块链行业内研究和开发的一小部分，工作组认为未来将有非常多的机会，从其他生态扩大EOSIO的应用场景以及获得更多知识。

整体而言，EOSIO生态系统需要吸纳融合除自身以外更广阔的区块链行业范围的一些举措，并为其做出贡献。这方面可以通过制定应用采用标准、提高不同生态系统的互操作性来实现。随着时间的推移，这些努力将减少企业、开发者和基于EOSIO的区块链构建的应用程序的最终用户之间的障碍。

## 核心原则

设计、构建和运作API的方法有很多。基于对EOSIO生态系统内需求的评估和研究，工作组希望在设计建立解决方案时，能够依据以下核心原则清单。



## 开发者优先

第一个原则是，确保高度关注那些将 EOSIO 集成到自身应用程序和服务中的开发人员的需求。满足这些需求的首要目标是留住现有开发人员，同时也需要吸引新开发人员加入生态系统。任何能够帮助开发人员更轻松地在EOSIO上构建的举措，都将会EOSIO带来更新鲜的创新和更有趣的应用场景。

## 前瞻性

在评估 EOSIO 内外开发人员的需求时，我们有机会思考未来可能存在的需求。在蓝皮书研究期间收集的信息，将尽可能确保这些系统功能可以在未来涉及 EOSIO API 的解决方案中使用。

## 支持去中心化

EOSIO 中每个组件的去中心化程度可能会体现在各方各面。某些组件可能获得或失去其可提供的价值，这取决于组件的不同处理方式。在设计解决方案的过程中，必须权衡不同方法的利弊，同时确保去中心化的最大可能性，并表现出对去中心化的偏爱。

## 可行性与实用性

在目前的EOSIO生态系统中，可用的人力和财政资源都是有限的。在提出包含解决方案的建议的时候，需要了解什么是可行的。提出的解决方案必须考虑现实，包括哪些是短期内可以执行的，哪些需要长期规划。

## 慎重规划

采用新的API解决方案将需要很多时间。所有使用API的应用程序和服务也都需要花费大量的时间和精力来更换或升级到新方法，因此推行新方法所需付出的代价也是巨大的。由于这些原因，未来的API解决方案的开发需要非常慎重地规划。

## 无障碍和低门槛

系统的设计还必须包括确保服务广泛使用的方法。使用EOSIO API的用户/开发者/团队要求数据可以用各种编程语言访问，可以从不同的环境中操作，并基于使用开源代码的记录标准。EOSIO API的设计不应该存在阻碍用户访问数据的因素，任何人都应该具备访问数据的能力和机会。

## 可互操作性

蓝皮书提案的解决方案应该涵盖推动 EOSIO 内部和外部的其他现有技术集成和工作的方法。对 EOSIO 所做的更改应该有助于推动增长 EOSIO 区块链在全球范围内的影响力，以及还应考虑将 EOSIO 技术与其他外部系统集成机会。EOSIO 社区也应该影响这些外部系统的发展方向。

## API 现状分析

在撰写蓝皮书时，工作组研究的主要部分涉及到了研究现有可用的 EOSIO API，探索在更广泛的区块链中的可行解决方案。在制定[提案](#)之前，了解我们目前的情况对于下一步计划非常重要。

为了理解本文的背景，需要对API概念进行一些解释。[维基百科的词条](#)描述为：

「应用程序编程接口 (application programming interface, 简称为API) 是[计算机](#)之间或[计算机程序](#)之间的连接。」

重申一下，API 允许您设备上运行的应用程序在连接到 Internet 时发送和接收信息。在本文中，我们通常将 API 视为能够与 EOSIO 数据交互的服务。

应用程序编程接口(API)是所有区块链的基础。用户需要能够提交包含在区块链中的信息，同时用户还需要能够查询区块链来接收信息。这两种活动，都需要区块链和用户应用程序之间的API。API是核心基础设施的一部分，必须按预期运行，才能满足最基本的操作要求。

EOSIO 生态系统中使用了几种不同类型的 API, 包括：

- 原生**EOSIO API**是EOSIO核心代码库的一部分，任何操作者只需设置一个配置选项就可以启用。原生EOSIO API是由Block.one和核心代码库的贡献者开发的。
- 社区开发的**EOSIO API**是需要额外配置和运行的应用程序，利用本地API以不同方式提供数据。这些API是由社区开发者作为独立项目的一部分开发的。
- 非**EOSIO**特定的**API**是已经开发出来旨在尝试将对区块链的访问标准化，同时保持独立于任何特定区块链的 API。EOSIO 目前尚不支持其中许多行业标准。

本节重点介绍原生和社区开发的 EOSIO API。为追求简洁，非EOSIO专用API仅在与具体[提案](#)相关的地方进行讨论。

## 原生 EOSIO API

EOSIO 目前有 4 个主要的原生 API 可供运营商使用: chain; history; state history 及 trace。

### Chain

chain API 是客户端读取与区块链当前状态相关的数据的方式。其中包括账户、代币和其他合约数据。chain API 中的 API 端口是在 2018 年 6 月推出时引入的, 随着时间的推移有一些微小的改进, 但结构没有重大变化。chain API 还允许客户端提交交易到区块链中。

### History

history API 是客户端如何获取已在区块链上执行的过去交易的有关信息。history API 的初始规范是在 2018 年 6 月发布时引入的, 但核心软件的原生插件已被弃用, 未经修改就无法正常运行。history API 的设计方式从未考虑过扩展问题, 导致其很快超出了其自身的限制。history API 被许多客户用来测试一个交易是否完成以及交易的后续行动。history API 还用于让客户检索特定账户的过去交易列表。

### State History

state history API 的开发是为了作为 history API 替换基础。state history API 提供了一些新功能, 例如流媒体和对区块链状态历史的专门访问等。state history API 于 2019 年 6 月在 1.8.0 版本中引入。state history API 允许其他应用程序使用历史信息并将其用于创建新的数据集和 API。

### Trace

trace API 是为了改善可扩展性问题, 过去许多开发人员社区已经强烈表达了对 API 历史解决方案缺乏可扩展性的不满。与 history API 或 state history API 相比, trace API 提供的 API 端口和数据可用性要少得多, 但确实也提供了一种更灵活的方式来存储和检索区块链中的历史信息。trace API 于 2020 年 3 月在 2.0.4 版本中引入。trace API 提供了一个端口让其他应用程序能够使用历史信息并将其用于实现各种目的。

## 社区开发的 EOSIO API

自 EOSIO 发布以来, 社区开发了许多外部 API 服务, 以响应开发者和运营商社区的部分需求。这些 API 都从上述一个或多个原生 API 中生成数据, 作为一个单独的过程来运作, 进一步丰富关联信息, 同时创建原生 API 中不易获得的有价值的信息。

## API 类型

### 衍生状态API

衍生状态API是指采用区块链数据的当前状态，并以原生API没有的方式对其进行扩展，用来满足特定的需求。衍生状态API可以用来满足特定合约或特定应用程序的需求。

### 流媒体API

区块链数据的流媒体是一个高需求的功能，需要立即访问相关数据的应用程序和服务。流媒体API被设计为既可以作为服务的中间件来检索自身数据，又可以作为需要特定实时数据的客户端的中间件。

### 交易历史记录API

尽管 EOSIO 具有多个与历史相关的 API，但可扩展性挑战以及对应用程序、API 和 SDK 的维护不足，导致社区开发了许多外部交易历史解决方案。

## 解决方案

原生的API一直嵌入节点系统中，被认为是官方版本。随着新API的需求产生，并且在Block.one明确表示不会开发解决方案应对新的需求的前提下，节点内部团队自我组织起来解决问题。

由此诞生的多种解决方案，虽然目标相同，但缺乏标准化，开发人员难以在各种解决方案之间自由迁移。而且这些解决方案的开发和运营大多是自筹资金开发并免费开放，并且依赖于开发人员和运营商的其他收入来源，这使得大多数开发团队难以确定工作的优先级并制定长期的计划。

截至 2022 年 2 月，社区开发的专用 API 被最多采用的包括：

	衍生状态API	流媒体API	交易历史记录API
Hyperion	✓	✓	✓
Chronicle		✓	
Light API	✓		
Firehose		✓	
dfuse	✓	✓	✓
Roborovski			✓

## Hyperion

由 EOS Rio 构建，基于state history API 开发，有效缓解了原始state history API 难以满足日益增长的使用需求的困难。Hyperion 是一个完整的历史解决方案，服务由原始 v1 标准逐渐成长并实施新的端口 (v2)。

Hyperion 是一个活跃的项目，不断更新并推出新功能。最近，Hyperion 开始允许为特定的合约检索创建插件，并被用于在 WAX 和 EOS 上提供 AtomicAssets API，以及在 Telos 上提供 EVM API 响应。

## Chronicle

Chronicle是由EOS Amsterdam基于state history API 开发的。该解决方案解析来自本地API的数据，并将其呈现为JSON数据流，供其他应用程序使用。

## Light API

Light API 由 EOS Amsterdam 构建，基于 Chronicle开发，而 Chronicle是基于state history API 开发的。其主要功能是提供有关 EOSIO 区块链账户和代币余额的信息。Light API被许多 Web 应用程序使用，包括 [Blocs](#)、[EOS Authority](#) 和 [Unicove](#)。

## Firehose

Firehose 由 EOS Canada(现为 StreamingFast)构建，并使用原生 EOSIO 软件的修改版本。这个修改后的 EOSIO 有一个额外的 API，用于实时传输事件流，称为「DMLOG」。Firehose 提供 gRPC API 及事件流供索引器和应用程序使用。区块流可以向后或向前播放，帮助了解分叉情况，并在断开连接时可以从同一地方恢复。最初的基于 EOSIO 构建的 Firehose 目前已兼容其他区块链并用于 The Graph。

## Dfuse

dfuse 在 Firehose 的基础之上添加了更丰富的过滤功能。此外，dfuse 还包括其他功能，包括在最后一个不可逆转的区块中存储每个账户的代币余额、在各种属性上检索交易、允许查询交易和状态、过滤所需的信息。dfuse功能更新不多。

## Roborovski

Roborovski 由 Greymass 团队基于 trace history API 开发。它从基于 trace history API 获取原始数据，并将数据以优化的格式存储，供将来使用。索引器能够以这种新格式为基础，并按照 ID 和账户提供交易信息。Roborovski 需要 EOSIO trace history API 的增强版本，目前尚未开源。

## 提案详情

在 API+ 工作组研究的过程中，我们采访了几十位经验丰富的区块链专业人士，去了解他们最大的需求和愿望。其中一些人指出，造成他们困扰的原因，一部分是由于缺乏知识造成的，在这种情况下，提供系统的、简单的指南教材就能够帮助他们的工作。但更多时候，他们所表达的需求超出了缺乏文件的范围，需要通过巨大的努力才能解决问题，满足需求。

对 EOSIO APIs 所依赖的系统进行战略性改进，可以大大减少开发者的工作。这些不同系统相互关联形成了一个复杂的网络，一个系统的任何变化都有可能对另一个系统产生影响。因此需要对这些系统进行必要的研究，采取全面的方法来得出有效的结论。

由于上述原因，目前提出的第一个提案是建立一个小组来协调研究和发展，持续推进工作。而且如何协调 API 架构的决定有可能会影响到其他每一个提案。

### [提案 1: API 研究和标准制定](#)

除了将更多关注度放在未满足的需求之外，已确定的许多需求也还有改进的机会。最常见的未满足的需求是蓝皮书中所有其他建议的基础。

- 开发已知的解决方案：
  - [提案 2: 交易执行周期](#)
- 支持网络运行的机会：
  - [提案 3: 专用 API](#)
  - [提案 4: 中心 API 服务](#)
  - [提案 5: 分布式 API](#)
  - [提案 6: 区块链数据库](#)
- 与现有和新兴行业标准的潜在集成：
  - [提案 7: Rosetta](#)
  - [提案 8: The Graph](#)

每个提案都可能提供多种解决方案。在计算人员总成本时，工作组将按照最高薪酬估算。当有几个方案时，每个方案都是独立存在，提出的总费用是该方案本身的费用。文中所有费用均以美元 (USD) 为单位。

## 提案 1: API调研和标准制定

在一套基本的API的基础之上, EOSIO的开发者和运营者建立应用程序。许多应用需求无法通过这些基本的API来满足, 开发人员需要更复杂的工具。

通常情况下, 开发人员可能需要多种复杂解决方案, 构建和维护应用程序的困难程度也随之不断增加。难度增加的同时造成了一种人为的障碍, 在没有大量时间和精力投入的情况下, 往往会阻碍EOSIO网络和应用的改进。

本项提案旨在通过对基本的API进行改进, 以及对一些更复杂的解决方案的组件进行标准化和改进, 来帮助开发者减少障碍, 减轻负担。

### 具体提案

建立一个API调研和标准制定团队(ARS团队), 协助建立下一代EOSIO API的开源软件。这是一项宏大的工作, 需要多个阶段的研究, 并协调EOSIO生态系统中一众团队共同开发、运行支持、持续维护该项目。

该团队将负责在以下方面协调软件生态系统:

1. 核心软件(nodeos)提供了哪些数据。
2. 这些数据如何提供给外部应用程序使用。
3. 标准化的外部API应该公开的数据格式。

其他的工作将在明确后逐步增加, 确保充分考虑整个EOSIO生态系统发展的需要。

### 需求清单

为了让大家了解该团队将负责帮助协调哪些类型的项目, 以下是工作组研究中, 一些已经非常明确的需求:

- **基础API:** 重新设计基本的JSON-RPC APIs, 优化调用模式并改善对关键数据的访问。实现REST模式并改进错误处理。
- **原生数据流:** 为跟踪/状态更改添加可选的原生数据流, 这与用户订阅的账户和合约有关。
- **交易执行周期:** 改进 API 处理交易提交及其执行周期的方式, 直到交易被包含在链上。
- **交易历史:** 标准化如何从核心软件中最好地检索交易历史信息, 以及客户的最终输出需要包含哪些内容。
- **API 可扩展性:** 创建一个框架, 应用程序可以使用该框架来构建合约或特定于应用程序的端点规范(例如 EVM、AtomicHub 等)。

- **API优化**: 允许横向扩展和只读数据访问, 减少存储要求, 增加功能和服务的可发现性, 并使其易于部署。

以上项目清单代表了多年工作的价值, 并还将随着其他需求的确定而继续扩大范围。

## 基本原理

这项工作需要在许多团队之间协调进行。以往, 各种EOSIO API组件的开发是以一种不协调的方式进行的, 这导致了对于不同需求的响应并不规范, 也造成了一些想要使用API解决问题的人缺乏明确的使用路径。

这些解决方案往往是闭门造车, 并且往往在没有考虑其他更广泛的潜在应用需求的情况下发布, 由此产生的解决方案受众很少, 并且缺乏对软件开发工具包(SDK)的支持。

## 工作职责

### 研究与评估

该团队将进行研究, 以便整理撰写一份全面的生态系统需求和愿望清单。这项工作可以通过与SDK和应用程序开发人员紧密合作来完成。对开源应用的评估和对API使用模式的观察, 也将突出可能值得关注的重点领域。

### 系统架构

通过研究过程收集的信息, 将在确定系统架构优先级的最佳路径方面发挥重要作用。该架构将确定每个部分彼此之间的位置、互相之间的依赖性以及如何更好的创建新的组件和全面的API解决方案。

EOSIO架构必须以一种易于理解的方式展示给开发者。这个架构将由关系图、API规范和设计目标等内容组成。文件的格式需要易于更新, 因为当利益相关者发现未满足的需求时, 团队必须能够迅速做出反应并采用组件来设计和提出新的解决方案。

架构的设计也应该考虑到每个组件的独特特性和其扩展能力。需要确定哪些是高需求的组件, 并确定其是否能够独立运行和横向扩展。

### 规范文档



每个定义的 API 组件都将附有一个规范文档，该文档详细描述了输入和输出的预期数据。这些规范将提供组件的每个单独实例必须采用的标准，确保相互依赖的各种组件之间的互操作性。这些提案将公布，并由开发者社区审查后发布。

## 协调发展

随着需求的不断确定，可能需要开发额外的 API 和软件。团队将协调开发工作，并确保为实现这些解决方案所做的工作符合整个系统架构。开发可以由任何一方完成，无论是通过自愿方式、奖金奖励、赠款方式或者是作为更大倡议的一部分来最终完成。

## 资助方式

### 为 ARS 团队提供资金

- 常设团队：ARS 团队可以得到一个初始预算以帮助团队能自主运作。另外还将设立一个多签账户，提供初始资金，确保该团队在特定时间内的运行。该团队应该组织起来，计划长期运作方式，外部实体来评估团队的表现并决定是否提供进一步的支持。
- 工作组：与 EOS 网络基金会赞助的工作组类似，ARS 团队可以得到预先确定的资金和限定时间表。该团队将进行研究，记录最佳的未来系统结构，并发布规范建议。然后，开发的协调工作将由不同的组织处理。
- 现有的团队：ARS 团队可以是一个现有的，或者是一个新成立的组织团队负责运作，比如一个同意承担该任务的基金会或公司。EOSIO 网络可以提供资助，补贴该组织的团队运行成本。

## 软件开发

- 预分配资金：ARS 团队可以获得初始资金池，以便在其认为合适的情况下，通过合约开发、奖金和捐助等方式为开发工作提供资金。如有必要，可向该资金池分配额外资金。
- 直接建议并获得资助：EOS 网络基金会可以采纳 ARS 团队的建议并直接资助开发。这可以在规定的时间周期内（比如每季度/每年）或以临时方式完成，具体取决于提案的数量。
- 公开发布建议并获得资助：ARS 团队可以公开发布其开发建议，这将允许来自各种 EOSIO 区块链的利益相关者提供资金或开发人员资源。

## 成本预估

此处列出的成本预估仅针对 ARS 团队，其他任何开发资金成本都将超出此范围，具体成本将取决于具体项目。

由于基于上述不同选项组建团队的方式具有灵活性，每个角色所需的实际成本将在本节中分解为各个要素。实际选择的团队成员数量将决定总体成本。

每个岗位角色都是以1年的成本来估计的，根据所提供的信息，可以转换更小的时间单位。

**系统架构师：年薪15万至25万美元。**

负责整体架构和单个组件的创建和调整。创建技术规范，了解所有组件的相互关联性，并致力于为外部项目/团队创建解决方案。与项目经理和技术作者协同工作。

**项目经理：年薪9万至11万美元。**

负责监督整个项目，与各种外部项目/团队和关键利益相关者进行沟通，并促进研究材料的收集与整理。与外部项目/团队、架构师和技术作者协同工作。

**技术相关撰稿人：年薪6万至8万美元。**

负责协助创建技术规范、标准、系统图和其他书面材料。与项目经理和系统架构师协同工作。

一个极简团队可以由一个架构师、一个项目经理、一个技术相关撰稿人组成，因为扩大架构师和技术相关撰稿人的数量，会加快开发的速度，所以我们也应该考虑建立更大的团队紧密合作来维护整个系统。

角色	极简团队	中等规模团队	人员充足的团队
系统架构师	1	2	3
项目经理	1	1	2
技术相关撰稿人	1	2	2
估计费用总额	<b>44万美元</b>	<b>77万美元</b>	<b>113万美元</b>

## 相关建议

工作组提出了以下建议：

- 从 Wallet+ 蓝皮书中 SDK 开发提案中获得的信息，将提供可能有助于未来 API 系统架构的内容。出于这个原因，团队认为 API 架构提案应该与 SDK 开发同时进行，或者在 SDK 项目完成之后进行。

- 该项目需要对系统具有非常充足的认知或能够快速了解情况的人领导。整个 API 基础设施的质量将取决于一个小型核心团队是否具有一致的愿景。因此，建议选择合适的人参与。
- 如果 EOSIO+ 工作组为 EOSIO 链之间的协作定义了一个整体结构，那么该项目在某种程度上应该属于其职权范围。
- 建议提供至少一年的中等规模团队资金预算。然而，根据招募团队成员的时间长短，第一年不太可能花费全部预算。

## 提案 2: 交易执行周期

EOSIO 用户体验中最关键的时刻之一是每当用户试图提交一个交易时，用户必须知道是否可以执行交易，以及交易何时能够成功完成。如今，许多应用程序和钱包都有实现这一点的技术，但这些相似的技术应该是所有 EOSIO 用户体验的核心部分。

作为 EOSIO 体验中无处不在的一部分，这些增强体验的功能应该直接应用于核心 API 软件，并通过软件无需分叉的升级提供给所有运营商。

### 具体提案

该提案旨在聘请 EOSIO 专家级开发者创建以下四个项目，解决核心软件中常见的用户体验问题。

#### 交易资源成本预估

由于目前在 EOSIO 的核心设计中采用了主观的计费系统，所以交易过程中的 CPU 或 NET 消耗的确切成本不可预计。然而，提交交易之前的成本估计可以直观了解交易是否会成功，以及本次交易大致会消耗多少资源，因此，我们将创建一个新的 API 端点，允许提交交易时，反馈资源使用成本估计。

#### 改进主观计费方式

在今天的 EOSIO 网络中，由于「主观计费」错误和节点之间的不同规则，点对点网络偶尔会出现交易丢失的现象。工作组将对涉及交易的代码进行研究，并评估更智能的引导交易的新方法。然后开发代码，以减少由于网络上个别节点的配置而无故丢失交易的现象。

#### 交易重写

为了帮助改善所有基于EOSIO的应用程序的用户体验，将在原生API软件中集成一个智能监控传入交易的系统。该系统将通过自身API跟踪接收到的交易，并确保交易在网络上中转。

## 交易最终状态

检查交易的最终结果，是专攻历史交易类API使用最多的功能之一。客户端在提交交易后往往会在后台查询确定交易是否成功。为了增加这些服务的可用性，并减少操作的复杂性，我们希望看到一个可选的功能集成到核心系统中，跟踪所有最近交易的基本信息和当前状态。

## 需要被考虑在内的影响因素

### 网络改进需求

在EOSIO这样的点对点网络上，出现不一致的网络情况并不奇怪，甚至许多项目都是专门为帮助缓解这种不一致而设计的。对于那些我们无法识别和提出解决方案的差异，工作组需要进一步调查研究。

例如，资源计费成本有时比实际成本高10-1000倍，这会导致网络中的交易失败，或者用户在交易过程中被过度计费。

诸如此类的问题也影响了系统的可靠性和在成本估计方面的准确性。我们建议在未来的提案中也对这些系统进行调查研究。

### 项目代码库

在管理这些功能的开发时，应遵循以下几个原则：

- 这些新功能应作为Mandel github repo协议的一个分支，并与「主要」分支合并。
- 需要添加单元测试和集成测试用例来涵盖新功能，并可能需要对现有测试用例进行修改。
- 代码的开发方式将允许社区提供反馈，因为功能已经达到了相关的程度。换句话说，在项目结束时不会有「代码大量下降」的情况出现。
- 截至目前，还没有计划建立一个由社区维护的开发者门户网站，所以文件将以git中的markdown文档的形式提供。这个文档将解释如何操作和使用所有的新功能，就像以前节点系统版本中的「发布说明」一样。

### 项目指南

需要 [ARS 团队](#)或其他精通 EOSIO 的开发人员来帮助指导这些功能的开发。

## 成本预估

目前的成本预估基于以下前提得出：经验丰富的核心 EOSIO 开发人员团队较为完整的实现了功能需求，包括适当的测试用例、自述文件等。

明细	费用
开发	38万美元
社区架构师/项目管理	2万美元

## 相关建议

工作组提出了以下建议：

- EOS网络基金会应该直接与具有实施这些功能的专业能力的公司签订合同。
- 这些功能应作为 Mandel 3.1 的一部分，显示节点系统开发的势头。
- 由于开发时间紧迫，以及ARS 团队尚未成立，API+ 工作组应监督开发工作的进展。

## 提案 3: 专用 API

正如蓝皮书[背景部分](#)所描述的情况，在EOS推出的几个月后，原生history API插件无法与区块链的交易量规模相匹配。这也是第一个示例，展现了开发者社区呼声非常高的必要功能，因此需要花费大量精力来解决核心软件的外部问题。

目前，开发和运营专用API基础设施的团队基本处于亏损状态，并且大多是通过出块奖励来补贴产品开发。开发资金的不稳定性已经显现，造成开发工作停滞不前，专用API产品非常不稳定。

## 具体提案

本提案中概述了几种支持专用 API 开发维护的方法。每个选项都提供一定财务支持承诺，来帮助未来几年专用 API 开发维护的可用性和稳定性。

## 软件维护

支持外部API软件的持续发展，帮助其适应EOSIO核心软件即将发生的变化。维护安全更新，与SDK开发者合作，更新文档，并提供开发者/运营商技术支持。

## 支持现有运营商

维护目前专用 API 软件和服务提供方，并确保其在至少未来 1 年内继续提供公共 API 服务。

在正式确定未来架构并投入开发之前，现有的应用程序仍将需要使用现有专用API。根据 [ARS 团队](#) 的建议，在新解决方案投入开发后的一段时间内，也将需要继续使用部分现有的专用API来过渡，尤其在无法保持向后兼容性的情况下。

因此，应该每年对目前专用 API 软件和服务提供方的支持情况重新做一次评估，来确定EOSIO生态系统中现有专用API的运行健康状况。

## 提高可用性

通过支持现有团队或为新团队提供机会，扩展生态API的运营并提升服务能力。利用现有团队将集中责任且能降低成本，而引入新团队将分散责任但却提高了成本。

## 功能开发

一般情况下，专用API的开发者会根据运营商/用户的反馈以及自身想法，来为自己的解决方案添加新功能。但在未来，新功能开发应该遵循[ARS团队](#)制定的统一标准。因此工作组建议，在标准化工作完成之前，先不要资助任何新功能的开发。

## 基本原理

### 用户体验风险

EOSIO网络的声誉将很大程度上取决于专用API是否可用。在专用API不可用的情况下，许多应用程序会收到无数的用户投诉，应用程序开发者无法控制局面。要吸引更多的开发者在EOSIO上构建应用程序，那么就需要保证应用程序所依赖的API正常运行，因此保证现有的应用程序正常运行是非常重要的。

### 目前所处形势

截至 2022 年 2 月, 许多 EOSIO 区块链的专用API情况都不太好。举个例子, EOS 网络上可供公众使用的专业服务如下:

- 2 个 Hyperion 实例(由 EOS Rio 和 Sw/eden 运营)
- 1 个 Roborovski 实例(由 Greymass 运营)
- 1 个 Firehose/dfuse 实例(由 EOS Nation 运营)
- 1 个 Light API 实例(由 EOS Amsterdam 运营)

EOS网络, 目前缺乏人才储备、能力提升及对专用API的兼容性。这些服务中的任何一个停止服务或停止开发, 都会对依赖它们运行的EOS应用程序构成潜在风险。当涉及到专用API解决方案时, EOSIO也存在许多类似的情况(注意:EOS上的公共API端口比此处列出的数量要多)。

## 需要被考虑在内的影响因素

### EOSIO 更新

在短期内, EOSIO平台会产生很多重大开发和变化, 这能够从[Mandel](#)硬分叉和[EVM+工作组](#)等项目中预见。因此工作组需要考虑这些重大变化对现有解决方案的影响。在计划时, 需要考虑为API解决方案提供基本维护之外的额外资金来应对开发变化。

### API架构的时间安排

在提出改进所有EOSIO系统提案的同时, 我们还必须考虑到其他新系统的规划、开发和部署所需要的时间。未来提出的解决方案可能需要相当长的时间才能投入开发, 并被网络广泛应用。由于许多EOSIO网络项目已经上线运行, 现有的基础设施需要维护的同时, 还要计划未来的升级时间安排。

### 缺少高质量的开发专家

目前所有API项目可用于开发的专家数量都有限。如果开发专家缺乏, 会造成重大的项目风险。如果社区希望增加开发人员储备, 预期单个项目的开发人员一年需要花费15万美元以上的费用。那么在EOSIO会有大量开发项目工作的前提下, 应该尽早拨付资金, 新开发人员需要几个月的时间才能够招聘到, 然后为项目做出开发贡献。

### 网络增长

关于这方面很难预测确切的运营成本, 因为运营成本高度依赖于网络使用情况。使用量增长会导致网络带宽和服务器CPU、RAM和存储空间需求增长。提供的估计成本可以根据运营商对于支

付购买、租赁或云计算成本的偏好。提案中使用的假设是网络增长率大致相当于迄今为止网络的平均增长率和使用率。如果产生显著增长，则应考虑额外的资金预算。

## 供应链问题

截至2022年2月，持续存在的供应链问题影响了全世界的计算机硬件供应。提前为运营商提供资金是明智之举，因为一些需要专用物理硬件的解决方案，需要至少6-9个月的提前准备。

## 对资助对象的工作要求

### 透明度

为了提高资助透明度，每个受资方需要至少每季度发布一篇关于项目进展的博客文章。文章应概述其工作细节。在今后向任何项目提供资金之前，向社区报告将是一项硬性要求。

### 代码开源

为外部 API 提供开发资助的前提是其必须做到代码开源。例如，支持 Roborovski 开发的前提是其作为开源产品发布。

## 成本预估

### 开发

在一个不断变化的区块链生态系统中，专用API需要持续的维护和开发。开发成本包括软件开发人员、技术相关撰稿人和项目经理的工资，以及其他成本。为了简洁起见，成本估计的详细分类省略。以下是针对1年开发的成本估计。

软件维护：这些费用包括专用API类型的API的基本维护和开发。这部分成本与软件开发人员为跟上即将到来的EOSIO变更、解决未决问题和其他微小改进所需的预期时间密切相关。

明细	费用
dfuse	7.5万美元
Firehose	7.5万美元
Light API	4万美元



Hyperion	15万美元
Roborovski	7.5万美元

该成本估计前提是 Mandel 3.1 不会对现有的节点系统接口进行重大更改。

## 运营

此处运营成本包括保持基础设施(服务器、网络)和运营商的平稳运行。为了简洁起见,省略估计个人成本的详细分类,但为了更清晰,硬件基础设施将分开进行成本估计。以下是针对1年运营的成本估计。

**支持现有的运营:**支持目前在EOS中运行的高使用频次的API,确保其稳定运营。该项成本估计将用于确保服务运营的最低金额,而不是运营所有解决方案的总成本。

**提高现有运营的能力:**提高现有API运营能力有关的成本估计。该项成本估计包含运营该解决方案所需的额外服务器和人员的费用。

**提高现有运营的能力储备:**为现有API增加运营能力储备的成本估计。该项成本估计包括在不同地方分配业务以增加运营弹性所需的新硬件和人力资源的综合成本。

现有运营	支持服务		新能力 *		新储备*	
	人力	基础设施	人力	基础设施	人力	基础设施
dfuse by EOS Nation	5万美元	1万美元	2万美元	2.5万美元	2万美元	7.5万美元
Firehose by EOS Nation	1万美元	1万美元	0.5万美元	2万美元	0.5万美元	2万美元
Light API by EOS Amsterdam	1万美元	0.5万美元	0.5万美元	1万美元	0.5万美元	1万美元
Hyperion by EOS Rio	5万美元	1万美元	2万美元	3万美元	2万美元	3万美元
Hyperion by Sw/eden	5万美元	1万美元	2万美元	3万美元	2万美元	3万美元
Roborovski by Greymass	5万美元	1万美元	0.5万美元	2.5万美元	0.5万美元	3.5万美元

\* 新能力和新储备的成本不包括在支持成本中。

新增团队: 招募能够推动专用API上线的专业开发者。该方面人才应该精通现有的EOSIO API操作。由于解决方案中存在不熟悉系统的学习成本, 使用现有团队会增加成本。现有团队将着重负责培训和指导解决方案的部署。

新增团队	培训	人力	基础设施
dfuse	1万美元	12万美元	7.5万美元
Firehose	0	2万美元	2万美元
Light API	0	4万美元	1万美元
Hyperion	\$5,000	7万美元	3万美元
Roborovski	\$5,000	7.5万美元	3.5万美元

## 相关建议

工作组提出了以下建议:

- 确保专业的 EOSIO 基础设施提供商和软件开发商长期发展, 为支持现有运营及软件维护提供资助。并应每年审查资助情况。
- 为扩大现有运营能力提供资助, 确保为网络增长做好准备。
- 招聘在 EOSIO 方面具有经验丰富的团队负责Firehose、Roborovski 和 Light API, 提高应用程序开发人员的弹性。在一般方案中每个解决提案将由2个团队负责, 在战略方案中由3个团队负责(dfuse除外)。

相关建议在「一般方案」中进行了总结:

明细	极简选项	一般选项	战略选项
软件维护	✓	✓	✓
支持现有运营	✓	✓	✓
提高现有运营的能力储备		✓	✓
最大化提高现有运营的能力储备			✓
新增团队 *			
↳ Firehose/Roborovski/Light API		+1	+2
↳ Hyperion/dfuse			+1

总成本	69万美元	111万美元	190万美元
-----	-------	--------	--------

\* 除了现有的团队外，增加新团队。

## 提案 4: 中心 API 服务

如今，在EOS中提供公共和专用API服务并不是一项可持续的业务。因为提供这些服务的成本较高，甚至超过了用户愿意支付的访问费用。

这种情况的一部分原因是运营和扩展公共及专用API服务成本很高，也因为应用程序必须通过API访问数据的无效方式。从长远来看，可以通过创建更高效的 EOSIO API 解决方案并降低操作的复杂性来改善情况。但尽管存在很多挑战，还是应该在短期内采取一些措施，为现在的开发人员提高服务的使用门槛。

虽然一些非EOSIO的解决方案提供了潜在的可持续性发展模型，但现在它们也都开始遇到EOSIO链固有的问题，例如处理、存储和服务具有数亿区块规模所需花费的大量区块链成本。

目前，在区块链行业中主要有两种解决方法。一是提供区块链即服务(BaaS)的中心化公司，如Infura和Getblocks，另一种是为开发者创建去中心化的生态激励模型，如The Graph。

### 具体提案

资助一个专门的团队提供API服务，并将其建设为公共资源，可以将新的开发者和企业带入生态系统。通过新的资金资助方式，减少公共服务对出块奖励的依赖。

### 基本原理

有少数出块节点通过使用区块奖励提供的资金来提供公共服务API。这些出块节点获得的区块奖励主要是为了促进出块运营，再将剩余资金投资于公共服务API等额外开发运营工作。这就造成了大多数公共服务API是在没有额外支持的，出块节点只能尽自身最大努力提供服务。

### 建议提供的服务

#### 公共产品

在上述背景下，建议对一些坚持在该领域的项目进行持续补贴，用于提供基本服务的公共产品。概括地说，包括以下内容：

- 网站/品牌建设：创建一个网站，欢迎感兴趣的人加入。
- 公共端口对接：为其他运营商提供可靠连接。
- 公共 API 访问：对基础 API 服务的访问进行速率限制。
- 公共下载：数据运营商需要开始运行自己的服务。
- 文档：运营商和开发人员的支持文档。
- 支持：在服务方面为运营商和开发者提供帮助。

## 付费服务

可以部署额外的付费服务，保证对资源的访问，满足客户的特定需求，并为企业提供更专业的支持。这可以通过为相关团队建立营利模式来完成，也可以让该团队作为更大组织的一部分。

免费服务可以作为进入专业服务的大门。

## 资金考虑

创建和资助该提案的潜在方法有很多，特别是 EOSIO 生态系统在 2021 年到 2022 年发生的飞速的变化。

## 管理结构

要对如何建立团队做出准确的估计，需要回答这个提案如何执行的一些基本问题。

- 领导：谁领导实施提案？
- 所有权：谁拥有品牌、网站、服务器和其他资产？
- 运营：哪个团队负责管理服务？
- 培训：哪个团队负责管理网站和文档？
- 支持：哪个团队负责帮助支持开发人员？

## 服务内容

服务也有大量的变量需要定义，由相关团队和网络用户的需求决定。服务对开发者越友好，提供服务的成本就越高。下面是几个应该考虑的示例：

- 地区：如果基础设施靠近终端用户，则他们的体验是最好的。因此我们需要明确它将服务哪些地区？在中国的访问可能需要额外注意。
- 语言：支持哪些语言？
- 支持：用户支持的时区覆盖范围是哪些？

- 功能:除了基本的API, 还提供哪些服务和功能?
- 部署:功能如何部署?支持哪些类型的服务器?(自有服务器、租用服务器、云服务等)
- 容量:目前需要多少资源, 下个月需要资源是否有变化?
- 开发环境:开发环境和生产环境都应该提供访问权限吗?
- 测试网:开发人员依赖测试网进行测试。如果有的话, 应该支持哪些测试网?
- 测试服务:随着新的API的开发, 是否应该提供给开发者进行测试?
- 开源软件:是否应该提供和支持为协助操作而开发的软件?

## 减少风险

在考虑如何处理解决方案的各种选项时, 必须认识到, 最快的部署速度将来自更中心化的方法。雇佣一个能够提供一站式解决方案的团队, 可能会很快实现交付, 但会产生对某个团队和对该一站式解决方案的过度依赖。

相反, 寻找那些租用服务器/数据中心部署服务的团队可能会延迟服务交付速度, 但有助于消除可能产生的单点的中心化故障。

所有这些问题解决方式的最终选择, 都会对所需的预算产生巨大的影响。

## 交付方式分析

可以通过多种方式交付解决方案。如果有计划的话, 交付方式甚至会随着时间而改变。下面概述了一些可能的情况, 其他变量也可能提供这些示例中未列出的好处。

	方案 1	方案 2	方案 3
领导	签约团队	中心实体团队	中心实体团队
所有权	签约团队	中心实体团队	中心实体团队
培训	签约团队	签约团队	中心实体团队
运营	签约团队	签约团队	中心实体团队
支持	签约团队	签约团队	中心实体团队

### 方案1: 将所有责任委托给签约团队

如果新的API运营被委托给一个现有的团队, 管理结构的问题都可以由一个现有的团队来应对。选择的现有团队可以是属于API+工作组的任何一个团队, 或者是这个领域中能够完成这项任务的其他团队。

服务产品将由该团队负责运营，并且定期、公开透明的向提供资金的利益相关者报告团队最新情况。

未来对外部团队的支持将采取问责制。如果该团队没有充分满足社区的需求，则以后提供的资金可以作为杠杆发生变化。这个方案的一个潜在缺点是，外部团队将拥有对产品及资产(品牌、服务器等)的所有权，如果该团队停止运营，则没有任何应对的方案。

## 方案2:网络负责领导，外部团队管理运营

管理结构的另一个选择是，由一个中心实体团队召集签约多个团队执行这些任务。这将允许一个中心实体团队保留对资产的所有权并负责领导，同时将所需的工作外包给外部团队。

需要实现哪些服务项目，可以由领导层来定义，而那些承包工作的团队可以提出建议。这个方案创造了一个责任层级，并确保特定网络的需求得到满足，但这非常依赖中心实体团队的责任感。

## 方案3:由网络领导和运营

与选项 1 相反，这个方案将通过一个中心实体团队解决所有管理结构问题。中心实体团队需要有工作人员来执行所有这些职能，并对资产和运营拥有完全的控制权。

## 成本预估

尽管确定最佳方法很复杂，但根据目前的条件，已经可以估计一些费用。实际的成本会因项目的每个方面的决策而产生变化。且有些费用高度依赖于所选择的方案的成本。

### 品牌塑造

服务提供商拥有强大的品牌塑造能力，不仅可以展示其形象，并对其成功产生正面影响。品牌塑造包括服务提供商的身份、品牌设计美学以及其实用性。这项工作的成本预估，低至基础设计的5万美元，高至精心设计的品牌标识和相关商标的20万美元不等。

### 组织搭建

服务提供商将需要一个法律实体组织，用以购买硬件并与服务提供商签订合同。通过使用现有组织提供此服务，可以避免产生该项成本。否则，启动成本估计将在3万至6万美元之间。

每年3万至4万美元范围内的会计和法律服务成本，也需要考虑在内。

任何组织都需要领导角色，例如 CEO、CFO 和 CIO。根据提供服务的复杂程度，这些角色可能是兼职或全职形式。

## 计费基础设施

如果提供付费服务，则需要计费基础设施。要构建计费基础设施，则需要组建一个由项目经理、前端开发人员、后端开发人员、智能合约开发人员和文档撰稿人组成的开发团队。构建计费基础设施的成本估计是 25 万至 35 万美元。

随着支持性服务的不断发展，开发团队每年需要规划 2.5 万美元至 5 万美元的维护费用。

## 部署方式

所提供的服务以及对应部署方式将因所选方法而不同。

**云计算：**所有的硬件都由云供应商托管，并使用一个单一点对点的平台。这是最快速的部署方式及最简单的管理方式。云计算提供了一个非常灵活的环境，可以满足 EOSIO 生态系统不断变化的需求。但所有优点都将伴随着巨大的成本，而且成本会随着使用量的增加而增长。

**专用租赁：**所有硬件由专用服务器租赁公司托管，并在服务器层面进行管理。虽然不像云计算那样可以快速部署和管理，但与自己购置持有硬件相比，专用租赁服务器所需的后勤服务更少。专用租赁的成本将比云计算低得多，但从长远来看，仍然比自己购置持有硬件的成本高。

**自己购置持有硬件：**购买硬件、租用数据中心的空间来创建专用的自持的资源。这种方式部署最慢，灵活性最低，并且需要签订长期服务合同。自己购置持有硬件的成本一开始会很高，然后从长远来看比云计算或专用租赁要便宜得多。

**混合模式：**在上述选项之外，还可以采用混合方法，优化管理、前期费用和持续性费用的成本。

无论选择哪种方案，都需要考虑创建及部署的额外成本，如部署自动化和监控工具等。

## 运营/支持

服务能够正常运行，需要员工管理基础设施、开具发票并提供客户支持。提供优质服务的成本会迅速增加，因此我们需要对此展开讨论，以确定实际可提供的服务水平。

**可用性：**应用程序开发人员最好能保证 7×24 小时的服务可用性。与工周一至周五的 9 点至 17 点支持模式相比，这种服务需要有大量的工作人员待命。维基百科上关于 [高可用性的词条](#) 提供了很好的说明。

语言：服务能够支持的语言种类，对成本也有很大影响。网站、文档、知识库和其他资产都需要多语言版本。客服服务人员也需要用不同语言回答技术问题。

## 相关建议

根据 API+ 工作组进行的研究，在此就如何更好的为网络创建服务提出一些建议。

## 确定方法

这个具有大量可能性的提案，很难以简单的方式呈现。为了在该提案上取得进展，工作组需要就如何采用中心化解决方案，以及如何对提供资金支持的网络负责等问题，做出最终的决定。因此本提案中没有提供具体的成本估算。工作组建议继续讨论，进一步完善该提案并制定出资金计划的最终理想版本。

## 规划前期投资

理想情况下，为服务提供资金最好是一项长期计划。包括数据中心和硬件租赁在内，运营成本一般在三年内分摊。托管提供商的服务器租赁也有租赁期，通常为1年。因此，根据该项目领导的方法，1至3年的资金支持规划更有利于资金的合理利用。

## 账户增长

无论是免费的还是付费的服务，随着用户需求的增长，成本都无法立即计算出来，需要领导层不断评估。随着网络的不断发展，在全球范围内提供迅速、及时的服务所需的额外资源，将增加成本估计。因此，应当预先预估设定一个合理的增长率，并纳入成本考虑中。

## 提案 5: 分布式 API

### 具体提案

这项提案旨在提供资金支持，用于编制一份白皮书。本白皮书将概述如何建立和运营一个去中心化的自治组织(DAO)，以激励EOS和其他基于EOSIO的多种类型的链上API的运营。

现有的已组建或计划组建的团队将运行其基础设施，而DAO将激励多个团队提供高可用性服务。



与[中心API服务](#)一样，这项提案的目的是创建高度可用的API基础设施。然而，其方法是非常不同的。这两项提案可以同时获得资助。

## 基本原理

如[专用APIs](#)所示，已经有五个团队在 EOS 上运行专用 APIs。很多团队为 EOS 提供原生 API 解决方案，而且还有更多的团队为其他基于 EOSIO 的链提供 API 和历史解决方案。这些团队拥有经验丰富的 EOSIO 运行工程师，也是其中一些解决方案的开发者。此外，团队来自世界各个地方，也存在地理和文化的多样性。

现有的EOS激励结构只对区块生产者(即BPs)进行奖励。由于这种激励结构不重视基础设施，所以很难吸引团队进行基础设施建设。因此需要有一种不同的方式组织所有团队协同工作，同时奖励团队已经提供的服务。

新的结构可以利用已经完成的投资和运营，来为生态提供高可用性、高度可扩展的API。

## 成本预估

生态将需要提供25万美元，这笔资金将分配给≤5个有丰富经验的EOSIO团队，用以合作撰写白皮书。在获得这笔初始资金之后，未来该项目团队将寻求其他资金来源，例如面向风险投资公司的私人代币销售或发行代币等。

## 相关建议

工作组提出了以下建议：

- 该提案没有包含有关 DAO 将如何工作的足够信息。如果社区对此提案感兴趣，大家可以聚集在一起并创建一个更详细的提案。根据详细的提案，可以分配资金来构建DAO白皮书，类似于最初的工作组的资助方式。
- 在花时间制定更详细的提案之前，需要收集更多来自社区的反馈，从而进一步确定这是一个值得研究并运作的项目。

## 提案 6: 区块链数据库

### 具体提案

这将是一项持续的资金支持计划，旨在支持一个或多个基础设施提供商，为任何需要使用的用户提供可下载的历史数据。该提案旨在正式确定此项支持计划，并让更多服务提供商/更多地区可以提供更加频繁的更新。

## 基本原理

对于想要运作自己基础架构的应用程序开发者来说，可供下载的历史数据非常重要。截止到目前为止，数据下载是由EOS中几个不同的出块节点提供的，每个节点的能力和更新频率各不相同。现有供应商包括：

- [EOS Amsterdam](#)
- [EOS Nation](#)
- [Greymass](#)
- [Sw/eden](#)

目前我们应为生态提供不同类型数据下载的可用目录。[bp.json](#) 是当前描述 API 可用性的方式，因此应扩展 bp.json 规范以支持不同类型数据下载。

同样，[EOS Nation Validator](#) 是团队查找 API 位置和其他相关资源的参考，也应该根据 bp.json 规范的更新进行同步更新，显示可供下载的可用数据。

在未来，ARS团队还应该规范数据下载格式以及包含与可用数据相关的元数据的API。这可能包括torrent或IPFS类型的下载。提供数据下载服务的团队也需要做出相应的调整。

受到资助的数据服务提供商，需要每季度公布一次下载指标，让大家可以了解具体情况。

## 成本预估

更新 bp.json 规范和验证器工具的成本很低。该项成本仅涉及短期开发人员和项目经理的费用。

项目经理：**0.5**万美元（一次性费用）

与社区合作，更新 bp.json 规范，以响应各种数据下载类型指定位置的需求。

软件开发人员：**1**万美元（一次性费用）

与项目经理合作，根据新的规范测试、更新验证器报告，并响应与更改相关的用户反馈。

另外，还需支持API开发人员从单个团队及单个位置下载数据。以下是已经运作EOS基础设施的团队的年度成本梳理。

快照:人力成本为每年1万美元;基础设施成本为每年0.5万美元。

应该每3小时提供一次新快照。当然也需要保持历史快照可用,频率可以减少到每周快照,这样可以节省存储空间。下载带宽应至少为1gb/s。

区块:人力成本为每年1万美元;基础设施成本为每年1.5万美元。

区块是一个非常大的文件。应该每周提供一个新的文件。且应该保留几个最近的副本。下载带宽应至少为10gb/s。

历史状态:人力成本为每年0.5万美元;基础设施成本为每年2万美元。

历史状态是一个非常大的文件。应该每周提供一个新文件。且应该保留几份最近的副本。下载带宽应至少为10gb/s。

历史追溯:人力成本为每年0.5万美元;基础设施成本为每年1.5万美元。

新的历史追溯可以分成不同的部分。这些新部分需要每天提供。下载带宽应至少为10gb/s。

**Firehose blocks:**人力成本为每年0.5万美元;基础设施成本为每年1.5万美元

每100个区块创建新的firehose区块文件。这些新的区块文件需要每天提供。下载带宽应至少为10gb/s。

可能的选项:

- 多个团队可以提供这种服务,其中一个可能是[中心API服务](#)。
- 更新的频率和数据副本的数量可以增加或减少(相应调整基础设施成本)。
- 数据下载位置的数量可以独立增加,而不用增加提供数据下载的团队数量(相应调整基础设施成本)。

成本预估影响因素:

- 该预算并不是一个团队运行EOSIO基础设施的全部费用,而只是提供数据的增量费用。且假定现有的EOSIO「专家」团队正在提供下载服务,因为预算不包括任何培训费用。
- 为了提高效率,「历史状态」、「历史追溯」和「firehose」的下载应该由提供区块下载的同一个人团队提供。如果不是的话,则需要增加费用。目前一个供应商不需要提供所有类型的数据。
- 如果节点系统开发改变了文件格式,导致新旧文件需要同时维护,则应考虑增加预算。
- 如果网络上的交易数量超过平均水平,导致下载文件变大,则应考虑增加预算。

	极简方案	一般方案	战略方案
更新 bp.json 规范和验证器	✓	✓	✓

快照	2	3	4
区块	2	3	4
历史状态	1	2	3
历史追溯	1	2	3
Firehose	1	2	3
总费用	16万美元	26.5万美元	37万美元

## 相关建议

工作组提出以下建议：

- 让一般方案的多个团队提供可下载的数据，且每个团队分布在不同地点，进行为期1年的服务。根据每年重新评估各团队具体工作情况，给予相应支持。
- 在「EOSIO」或「EOS网络基金会」的主网站上应该建立页面，列出可用的下载位置。目前这个选项的成本还没有计算出来，因为到现在为止还未确定最终管理方案。

## 提案 7: Rosetta

Rosetta 是一种越来越流行的 API 规范和工具集，它使与区块链的集成更简单、更快、更可靠。为EOSIO创建 Rosetta 将使 EOSIO 项目代币上线交易所变得更加容易、更加合规。

目前，交易所必须构建自己的中间件连接到 EOSIO 网络，这意味着交易所技术团队必须首先对协议有更深入的了解，然后再开发内部解决方案。官方的 Rosetta 可以大大降低集成 EOSIO 代币的成本。

该工作流程的关键是为EOSIO提供一个Rosetta可交付成果，供更多的交易所使用，最大限度地提高基于EOSIO的代币的流动性。合理的步骤是：首先了解交易所的要求，创建一份全面的技术文件，并提供承包商选择标准。对于这一初始阶段的建议是，让一批经验丰富的运营商和开发商参与和交易所的会议，并编写技术规范。

该解决方案的初步技术评估提出了一些需要社区验证的问题，特别是对于打算使用该解决方案的交易所来说，但考虑到设计决策不应开发工作产生影响。也就是说，这一解决方案是作为中间件存在的并嵌入nodeos，实现索引和查询历史数据、搜索交易和事件以及审计/合规性等功能。

## 路线图

路线图可以使各个工作阶段独立开来, 以便EOS网络基金会在需求变得更加明确、对每个交付产品的技术和合规要求建立更清晰认知时, 做出更合理决策。

### 1. 技术规范(8周)

- a. 与交易所和其他主要利益相关者举行会谈, 了解他们的要求。
- b. 验证关键设计决策并编写技术规范。
- c. 讨论并提出未来的结构建议。

### 2. 数据 API + 构建 API 基础(16 周)

Rosetta 通常分为用于从区块链检索数据的数据 API、用于构建和提交交易的架构 API 以及为交易和搜索实现额外端口以支持额外集成的索引 API。在第一阶段, 实施将侧重于基础构建。

### 3. 索引端口(8周)

可以分配额外开发时间以实现此功能。

### 4. 文档(4周)

为运营商和用户创建文档, 用于配合EOSIO Rosetta的实施。

### 5. 开发商/交易所支持(交付后12个月)

### 6. 审计: 可能需要审计, 但未列入预算。

## 成本预估

阶段划分	时间	费用
技术规范	8 周	2.5万美元
数据API + 架构API	16 周	7.5万美元
索引器终端	8 周	3.5万美元
文件编制	4 周	0.75万美元
开发商/交易所支持	1 年	2.25万美元
总费用		16.5万美元

## 相关建议

资助技术规范并承诺根据反馈为开发提供资金。应通过交易所和社区验证对开发人员/交易所支持的需求。为建立技术规范的团队提供资助，并承诺根据反馈为开发提供资金。交易所和开发者的需求需要经过社区交流验证。

## 提案 8: The Graph

### 具体提案

该提案旨在资助一个团队，基于 Firehose 展开调查，推动 EOSIO 实现支持 The Graph。在该项目执行结束后，EOSIO 应用程序开发人员将能够为基于 EOSIO 的链创建子图(图论的基本概念之一，指节点集和边集分别是某一图的节点集的子集和边集的子集的图)，并将创建几个示例子图，提供文档展示如何创建基于 EOSIO 的子图。开发人员可以通过 telegram 及 discord 获得支持。

Graph Node 已经普遍支持从 Firehose 中读取数据，但是目前还不支持 EOSIO Firehose。该提议旨在使 EOSIO Firehose 与其他区块链保持同步，并将 EOSIO 支持添加到 Graph Node 中。

### 基本原理

最初，The Graph 只支持基于以太坊的链。自 2021 年中期以来，The Graph 一直在努力扩展，开始支持 NEAR 和 Solana 等基于 Firehose 技术的项目，预计对其他协议的支持将继续被添加到 The Graph 中。

The Graph 有可能成为普遍存在的协议。因此 EOSIO 也不应该被落下。

开发人员能够以编程方式创建一个清单，该清单提供有关子图的数据源、模板和一些元数据的信息，并定义一个存储数据的模式，指明如何查询子图。

以太坊和其他 EVM 兼容链上的一些大型 DeFi 应用程序，利用 The Graph 为用户可视化的提供丰富的数据，协助用户获得深度数据图表(例如：代币价格、24 小时交易量、TVL 等)。

一些大的 DEX，如 [PancakeSwap](#)、[SushiSwap](#)、[Synthetix](#)、[Uniswap V3](#)、[CurveFi](#) 持有大量 GRT 代币吸引优质索引工具开发。EOSIO 项目同样需要提供兼容的接口。

DeFi 分析平台和运营方更喜欢使用子图，原因如下：

- 检索数据的API和工具在所有子图之间一致。
- 直接从链上区块提取的数据非常可靠。
- 子图是防篡改的。

## 需要被考虑在内的影响因素

- 该项目依赖于 Firehose 的维护和运营资金。
- 假设承担这个项目的团队需要学习这个项目 (如EOSIO、The Firehose和The Graph) 等相关技术。那么如果一个团队拥有这些项目相关的专业知识, 项目成本将会降低, 交付时间也会缩短。
- The Graph 在不断发展。在未来的几年里, 应该为这个项目持续提供资助, 因为需要不断的工作来跟上未来的技术变化。
- 如果将来, The Graph 被做为许多基于 EOSIO 的项目中使用的核心组件, 则应考虑在未来提供额外的资金, 以展示EOSIO功能的方式发展The Graph。
- Graph 基金会可能会提供资金来添加额外的技术堆栈, 如 EOSIO。可以尝试考虑研究与 Graph 基金会建立正式合作伙伴关系并探讨获取资助的方式等。

## 成本预估

由于该项目投资比较灵活, 因此每个角色所需的实际成本在本节中被分解开来。所选择的实际团队成员的数量将决定总体成本。

每个预期角色的估计成本以 1 年为单位, 可以根据所提供的信息预估更短时间的成本。

高级软件开发人员: 年薪**10万至15万**美元。

与例如 Defi Applications 和 The Graph 核心团队合作, 负责研究和开发解决方案。与项目经理和技术相关撰稿人协同工作。通过telegram及 discord为运营商和子图开发人员提供技术支持。需要精通 Go(Firehose)和 Rust(Graph Node)编程语言。

软件开发人员: 年薪**7万至10万**美元。

负责开发实例子图。接受项目经理和高级软件开发人员的任务。为运营人员和子图开发人员提供技术支持。

项目经理: 年薪**9万至11万**美元。

负责监督整个项目，与各外部实体和主要利益相关者进行沟通，并促进研究材料的收集。与外部实体、开发人员和技术出身的撰稿人协同工作。

技术相关撰稿人：年薪6万至8万美元。

负责协助创建运营人员和开发人员文档，以及其他形式的书面沟通材料。与项目经理和开发人员协同工作。

除了人力费用，成本还应包括用于开发和提供实例子图供公众使用的项目运行基础设施费用。成本估计为每年2万美元。

以下几个选项：

	经验丰富的团队 一般方案	经验较少的团队 一般方案	战略方案
基础设施	✓	✓	✓
核心软件开发	0.25	0.5	0.5
子图软件开发	0.25	0.5	1
项目管理	0.1	0.2	0.3
文档	0.2	0.3	0.5
总费用	<b>10.95万美元</b>	<b>19.1万美元</b>	<b>26.8万美元</b>

## 相关建议

工作组提出以下建议：

- 研究与 The Graph 基金会建立官方合作伙伴关系。
- 只有少数开发人员具备 EOSIO、Firehose 和 The Graph 的综合知识。通过为经验较少的团队提供适度投资，以探索基于 EOSIO 的区块链如何与 The Graph 协同工作，扩大基于 EOSIO 的开发人员队伍。

## 结语



在前面的章节中，我们只浅显的表述了针对项目的研究。我们希望通过这些信息，可以让社区达成共识，我们需要维持EOSIO生态系统中较为脆弱的部分能够稳定运作，并迅速采取行动推进现代化升级改造。

工作组建议采取以下行动：

1. 确定[ARS团队](#)在EOSIO生态系统中的位置，并建立一个能够实现未来目标的团队。
2. 赞助开发核心软件内交易执行周期的改进，改善用户体验并帮助开发人员构建应用程序。
3. 支持EOS系统中，目前应用程序所依赖的公共专用API稳定发展。
4. 决定如何运行中心API服务，为用户和在EOS上运行的项目提供访问权限。
5. 研究分布式API基础设施的可行性及其经济激励方式。
6. 赞助各种团队构建基础设施运营商矩阵以快速启动所需的关键区块链数据。
7. 启动探索EOSIO的Rosetta兼容性，从而进一步推动适配行业发展标准。
8. 开发软件推动The Graph协议与EOSIO集成，从而支持创建新的数据集。

下表列出了每个提案的成本估计明细：

提案	费用
1: API研究和标准制定	77万美元
2: 交易执行周期	40万美元
3: 专用 API	111万美元
4: 中心API服务	待定
5: 分布式API	25万美元
6: 区块链数据库	26.5万美元
7: Rosetta	16.5万美元
8: The Graph	19.1万美元

上述所有提案成本估计都需要具体讨论，所以欢迎大家给予我们反馈建议和意见：

- 我们概述的每项举措的正常投资标准是多少？
- 未来的研究应该侧重哪些点？
- 我们有没有遗漏什么？
- 我们需要考虑哪些新出现的趋势？

我们建议，研究应该无限期地继续下去，并定期撰写类似的蓝皮书。本篇蓝皮书可以算作第一版。这样的话，EOSIO内外的生态系统都会有一个最新的视角来审视发展方向。

# 致谢

API+ 工作组由来自这三个团队的才华横溢、资深的 EOSIO 开发人员组成。

- **EOS Nation**: EOS Nation是[Pomelo](#)、[.gems](#)、[EOS Name Service](#)、[EOSX 区块链浏览器](#)及[EOS Detective](#)的开发团队，也是目前排名第一的区块生产者。另外，EOS Nation 还是[dfuse on EOS](#)的基础设施提供商，负责维护其EOSIO代码库。
- **EOS Rio**: EOS创世区块生产者，[Hyperion History](#)的核心开发团队，Hyperion History是一个完整的解决方案，为EOSIO提供优化的数据结构和存储方法。
- **Greymass**: Greymass 是许多EOSIO区块链的创世区块生产商，致力于促进分布式账本技术和其基础设施的发展。Greymass也是[Anchor Wallet](#)、[Fuel](#)、[Unicove](#)、[EOSIO签名请求规范\(ESR\)](#)以及许多[核心网络和移动应用程序SDK](#)的开发团队。