

当前开发阶段：项目交付

预计开发时间：2020-04 至 2020-05

实际开发时间：2020-04 至 2020-05

开发内容简要说明：

借助区块链实现了电子证据的存储和固定。系统主要分为两部分：存证、验证。存证部分提供了两种存证方案，验证部分也提供了相对应的两种验证方案。经系统存储的电子证据能提高可信度，降低司法诉讼中验证电子证据真实性的成本，提高诉讼效率。

项目开发代码库：

<https://github.com/hdm926/aelf-boilerplate>

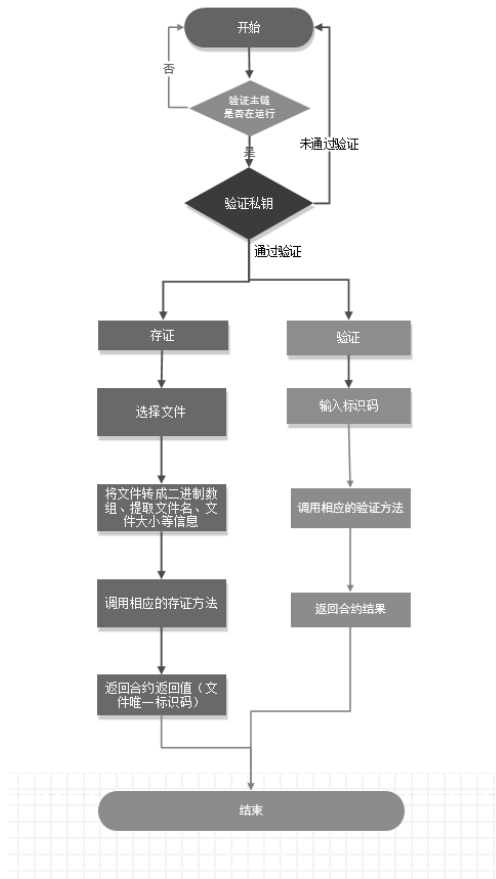
项目开发文档（需根据项目申请 pr 完成该文档）：

项目需求说明：描述项目的开发背景、用户、需求、功能等；

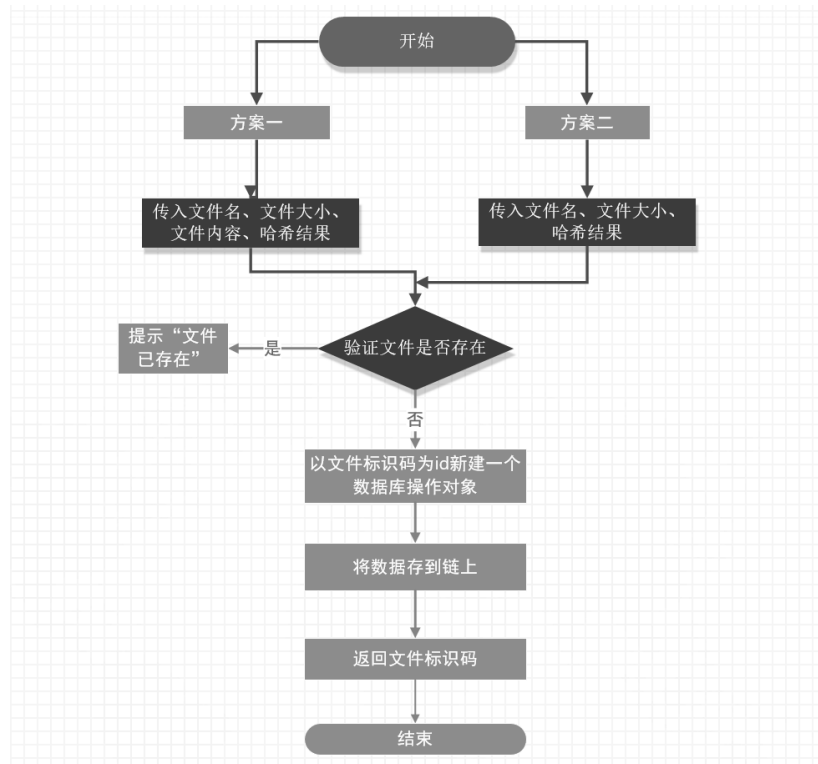
- 开发背景：随着信息技术的发展，电子证据在诉讼过程出现的频率高、范围广、种类多，由于易篡改，如何认定电子证据的真实性成了案件的首要难题；此外，电子证据也易消亡，通常需要多次备份存证且可信度的高低也依赖于原始存储媒介；再者，传统存证方式耗时久、效率低、成本高，也不适合电子证据的存证。在电子证据的生成、收集、传输、存储的全生命周期中，区块链可以对电子数据进行安全防护、防止篡改、并进行数据操作的审计留痕，从而为相关机构审查提供有效手段。本项目借助区块链实现了对电子证据的存证和验证操作。
- 面向用户：需要将各类电子证据等作为诉讼证据的用户；需要将自己的艺术创作进行原创声明以备日后被抄袭之后用于维权的用户；需要在诉讼阶段查看当事人提交的电子证据的司法机关；
- 需求：1) 对用户提供的电子证据进行哈希计算之后存储并固定；2) 根据文件标识码查看存储在链上的原文件信息。
- 功能：提供两种存证方案，方案一：存储文件哈希、文件二进制数组、文件名、文件大小等信息；方案二：存储文件哈希、文件名、文件大小等信息，不存储文件内容；存证完成后将文件标识码返回给用户；提供相对应的两种验证方案，验证方案一：用户输入存证时返回的文件标识码，系统根据该标识码返回存储在链上的文件二进制数组，并将数组转成图片之后正在前端界面显示；验证方案二：用户输入标识码，同时上传原文件，系统对原文件进行相同的哈希计算，并和链上存储的文件哈希进行比较，返回一个布尔值，表示文件是否被篡改过。

项目设计说明：

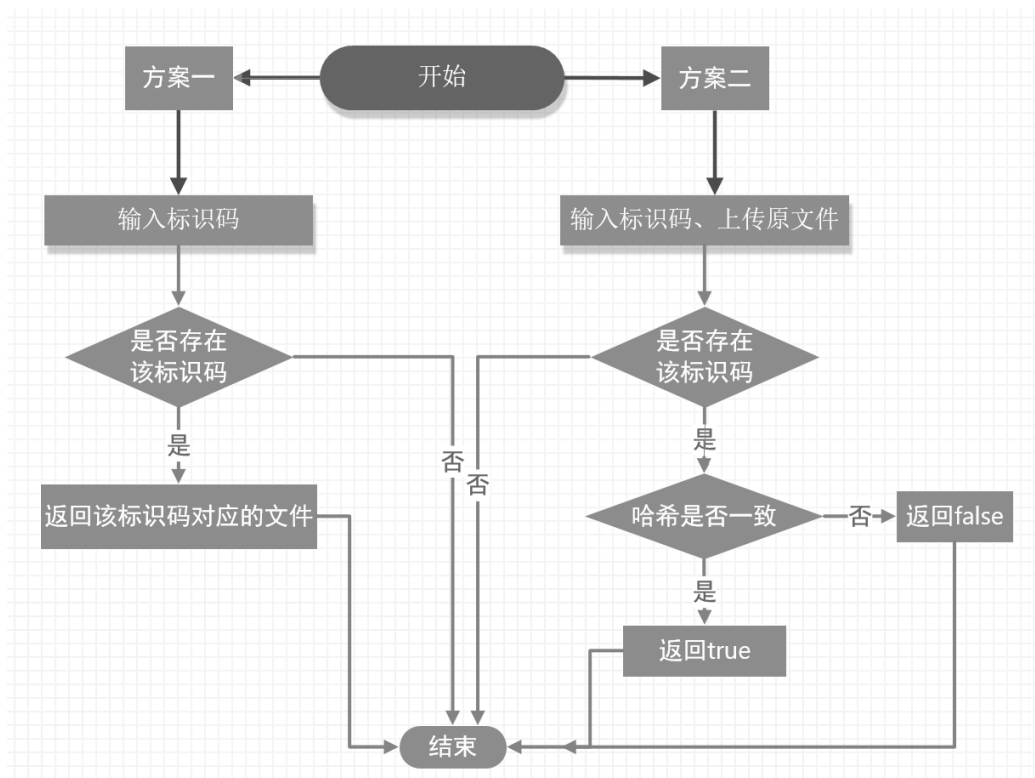
- 描述项目的框架设计：下图依次为系统流程图、存证流程图、验证流程图



系统流程图



存证流程图



验证流程图

- 数据库设计：下表中的两个对数据库的操作分别对应两个存证方案。

方法
MappedState<Hash , FileReceived> FileReceived{get;set;}
MappedState<Hash , FileReceivedPlanB> FileReceivedPlanB{get;set;}

数据库设计

属性名	数据类型	备注
id	aelf.hash	文件唯一标识码
fileName	string	文件名
fileByte	bytes	文件的二进制数组
fileSize	sint32	文件大小

FileReceived

属性名	数据类型	备注
id	aelf.hash	文件唯一标识码
fileName	string	文件名
fileSize	sint32	文件大小

FileReceivedPlanB

测试报告：

- 项目的开发完成情况：功能开发完成，已完成各项测试
- 环境：（1）合约使用 C#语言，编码 IDE 为 JetBrains Rider；（2）前端界面使用 JavaScript 语言,react 框架,ant-design 组件库,编码 IDE 为 WebStorm；（3）系统环境：Windows10 家庭版。

●

使用说明：

- 存证如下图所示：
上传的文件来自用户本地存储。两个按钮：“存图”表示系统将存储图片的全部信息；“存哈希”表示系统将存储文件哈希，不会存储全部文件内容。界面中央的哈希码为存证完成时返回的文件标识码。界面最下方的图片是用户选择上传的图片的预览。

732f73e5814762c3ffa807d41071428babfde8f21fdd539a29ec32ac414bea7c



- 验证
(1) 验证方案一对应存证时的“存图”，只需要输入存证时返回的文件标识码，点击“验哈希”，系统会将原文件显示在界面上。

浏览...

732f73e5814762c3ffa807d41071428babfde8f21 added539a29ec32ac414bea7c

验图与哈希

验哈希



- (2) 验证方案二对应存证时的“存哈希”，输入文件标识码，同时上传原文件，点击“验图与哈希”，系统返回比对结果：“哈希一致”表示图片未经篡改，“哈希不一致”表示此时上传的文件和存证时的文件不符，“标识码错误”表示链上没有存储该标识码对应的文件信息，可能是输入错误。

C:\Users\86136\Pictures\Saved Pictures\PIC10.jpg

浏览...

732f73e5814762c3ffa807d41071428babfde8f21 added539a29ec32ac414bea7c

验图与哈希

此站点提示...

哈希一致

确定

	输入	预期结果	是否符合预期
验证哈希	732f73e5814762c3ffa807d41071428babfde8f21fdd539a29ec32ac414bea7c	返回原文件	是
	632f73e5814762c3ffa807d41071428babfde8f21fdd539a29ec32ac414bea7c	提示“标识码有误”	是
验证图和哈希	(1)732f73e5814762c3ffa807d41071428babfde8f21fdd539a29ec32ac414bea7c (2)原图 PIC10.jpg	提示“哈希一致”	是
	(1)6bda8d23877675148c62cc0e073e859316e152cd3f180affdef7e4b87e9cb2a4 (2)原图 PIC10.jpg	提示“标识码错误”	是
	(1)7bda8d23877675148c62cc0e073e859316e152cd3f180affdef7e4b87e9cb2a5 (2)图 PIC11.jpg	提示“哈希不一致”	是

针对上图中的图片（图片名：PIC10. jpg），测试用例如下：

测试用例