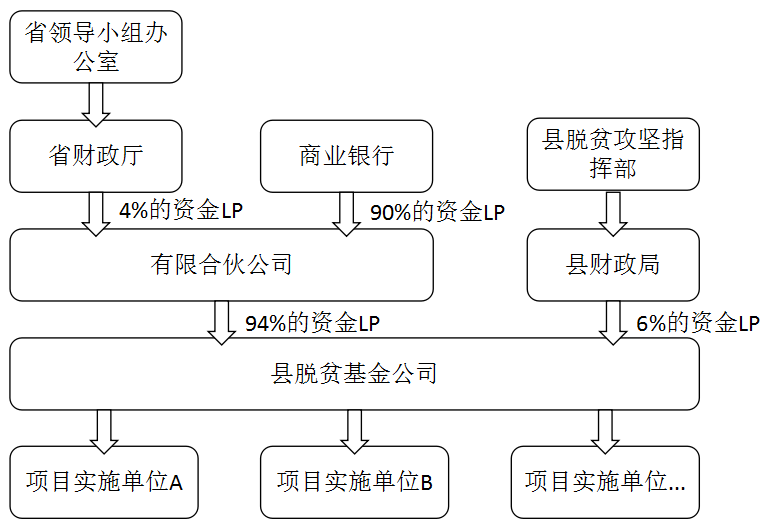
3.1 业务介绍

3.1.1 业务简介

按照《贵州脱贫攻坚投资基金设立方案》，设立贵州脱贫攻坚投资基金扶贫产业子基金（以下简称“产业基金”）。产业基金遵循“政府主导、企业主体、市场运作、风险可控”的原则，按照“强龙头、创品牌、带农户”的思路组织实施。政府通过财政出资、明确投资方向、界定投资范围、提供分险增信服务等方式发挥主导作用。

3.1.2 主要流程

业务参与主体如下图所示：



1. 资金募集

产业基金总规模3000亿元，财政性资金和金融机构按10%:90%比例出资，即：财政性资金出资300亿元，银行等金融机构出资2700亿元。其中，财政性资金省和市县出资比例为：4:6，即省财政厅资金出资120亿元，20个极贫乡镇所在县市的财政局出资180亿元。

产业基金采取募投制，投资项目成熟一个投放一个。为提高资金使用效率，避免募集资金闲置，降低融资成本，单个投资项目获批后由脱贫基金公司发起募集，其对应的10%财政募集资金到位后，90%的金融机构募集资金同步到位。

1. 资金投放

产业基金的投资方向为带动农民脱贫增收的产业，包括山地特色高效农业、林业产业、加工业、农业服务业、农产品流通业、农业信息化、旅游业、大健康产业、农村互联网产业、能够促进农村一二三产业融合发展的产业等方面。

1. 资金管理

产业基金投资由贵州脱贫攻坚投资基金管理领导小组办公室（以下简称“省领导小组办公室”）根据分期募资、按需使用的原则，实行限额管理、动态调剂。除贵阳市、贵安新区按切块方式外，其余市（州）按照建档立卡贫困人口数、贫困发生率等因素设定投资限额，市（州）与县（市、区、特区）比例为2:8。

市（州）、县（市、区、特区）级人民政府分别委托所属的国有企业（以下简称“受托企业”）履行本级产业基金出资人职责和义务，直接投资本区域内的企业，有限合伙企业按比例跟进投资。受托企业负责组织本区域内具体投资项目的申报遴选、投后管理和退出等。

20个极贫乡镇所在县市的脱贫攻坚指挥部组织项目申报，并纳入申报项目库。

金融机构会同受托企业对申报项目进行风控审核和遴选，并完成风控流程。金融机构对项目进行信用风险判断；受托企业对项目进行政策把关及专业判断。遴选出的项目归入省级遴选项目库。

金融机构否决的项目，经担保公司担保或其他增信方式选择项目作出增信后，满足金融机构审批条件的，可纳入省级遴选项目库。

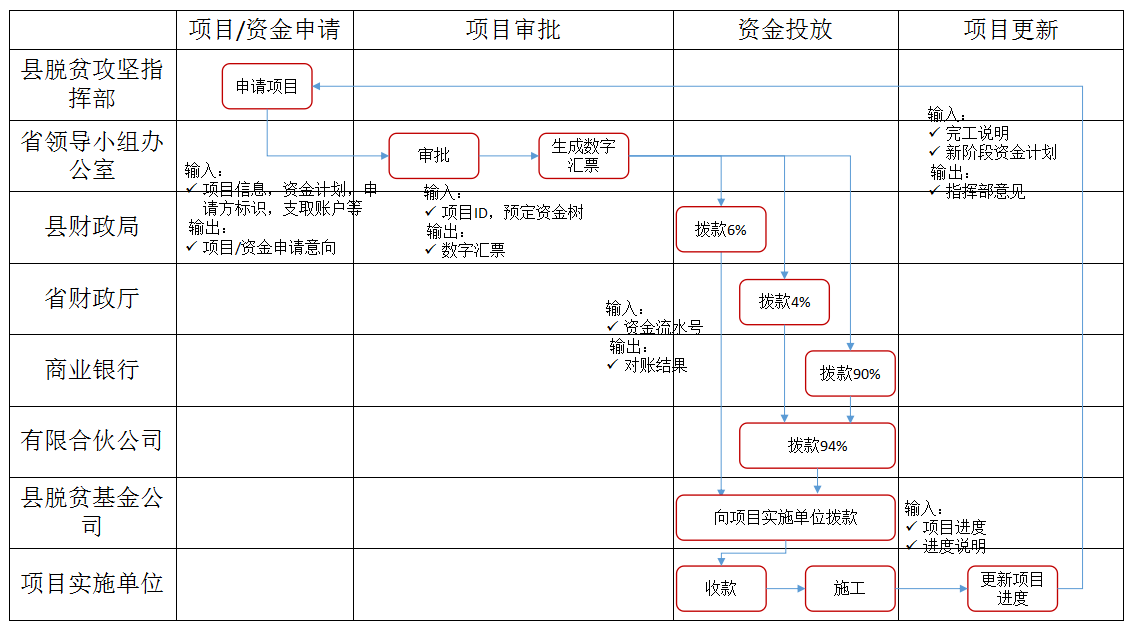
按照分级管理的原则，省、市级对遴选项目库中的项目进行审核后，由各级受托企业按程序组织对省级遴选项目库中的本级项目进行投资决策。

受托企业设立投资决策委员会，对纳入省级遴选项目库中的本级项目投资进行决策。投资决策委员会组成形式与决策程序由受托企业与有限合伙企业出资人协商确定。

省领导小组办公室牵头建立“贵州脱贫攻坚投资基金大数据业务管理系统”，对项目库建设、投资进度、资金使用、项目实施及投资退出实施信息化管理。省发改委、省财政厅、省农委、省扶贫办、省经信委、省商务厅、省林业厅、省旅发委等行业主管部门加强基金投资的行业指导、协调和培训。

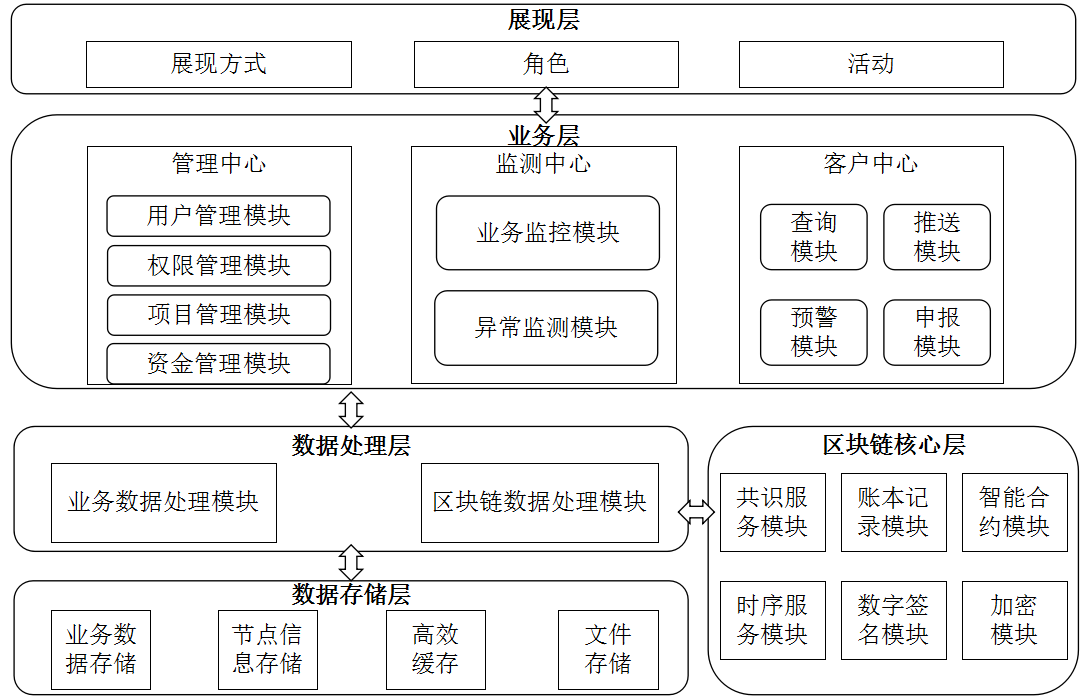
省领导小组办公室会同省级有关行业主管部门建立基金运行评价考核机制，设置定性和定量指标，考核结果定期向贵州脱贫攻坚投资基金管理领导小组报告。

相关流程如下泳道图所示：



3.2 技术架构

系统架构图如下：



展现层：

1. 展现方式：基于区块链的脱贫攻坚应用管理平台以web应用形式呈现。
2. 角色：角色有管理员、贵州脱贫攻坚投资基金管理领导小组办公室、省财政厅、县脱贫攻坚指挥部、县财政局、商业银行、有限合伙公司、县脱贫基金公司，项目实施单位。共计9种角色。
3. 活动：活动有身份注册与认证、项目申请、项目审批、资金拨付、项目进度查询、资金极度查询、项目进度更新等活动

业务层：

1. 管理中心
   1. 用户管理：处理处理LP，GP和项目实施单位等用户的加入/退出区块链系统。
   2. 权限管理：对角色所拥有的权限进行管理。
   3. 项目管理：对项目信息进行管理。
   4. 资金管理：对资金的申请、拨付、使用等进行管理。
2. 监控中心
   1. 业务监控：监控业务流程中的用户行为。
   2. 异常监控：监测项目或者资金的异常信息。
3. 客户中心
   1. 查询：LP，GP和项目实施单位根据自己的权限查询响应的信息，包括项目进度，资金流向等。
   2. 推送：LP，GP和项目实施单位个性化订制信息推送，免去查询环节，订制推送信息一有变化，第一时间向其推送。
   3. 预警：项目、资金如发生异常，发送系统预警。
   4. 申报：对发现存在的且系统未预警的问题进行申报。

数据处理层：

1. 业务数据模块：对业务层产生的数据进行处理和转换，如果需要存储的数据输出到数据存储层，如果需要上链的数据输出到区块链数据处理模块，并根据数据流进行负载均衡。
2. 区块链数据处理模块：对输入的数据按照区块链和智能合约的要求进行处理和转换并输出到区块链系统中。

数据存储层：

1. 系统数据存储：存储平台系统数据，包括用户信息、区块链节点信息等。
2. 节点信息存储：存储区块链网络节点即部署信息。
3. 文件存储：存储平台所有文件信息数据，包括大型文本文件、视频文件、音像文件等非结构化的数据。
4. 高效缓存：应对超高业务流量对系统的冲击。

区块链层：

1. 共识服务：区块链网络中各节点对在区块链系统中进行事务或状态的验证、记录、修改等行为达成一致确认的方法。
2. 账本记录：负责区块间P2P通讯以及分布式存储。
3. 智能合约：智能合约是一套以计算机代码形式定义的承诺，以及合约参与方可执行承诺的协议，即：用计算机代码形式编写合约参与方达成的条件型协议，当条件被触发时区块链系统自动执行该协议。根据应用场景的不同需求，区块链系统可有选择性地提供智能合约功能。
4. 时序服务：对于区块链系统中的行为或数据需记录相应的一致性的时序。
5. 数字签名：数字签名功能被接收者用以确认数据单元的完整性以及不可伪造性，即确定消息确实是由签发方签署的。
6. 加密：加密功能一般具体包括加密和解密两个操作：加密操作是把明文数据转化为密文数据，解密操作是把密文数据还原为明文数据。

3.3 技术路线

**基于区块链的脱贫攻坚应用管理平台**以web应用形式呈现，创新性的采用MVCB的软件架构。

MVCB即Model-View-Controller-Blockchain，即把每一个应用的输入、处理、输出流程按照Model、View、Controller、Blockchain的方式进行分离，这样应用被分成四个层——模型层、视图层、控制层、区块链层。

视图(View)代表用户交互界面，对于Web应用来说，可以概括为HTML界面，但有可能为XHTML、XML和Applet。随着应用的复杂性和规模性，界面的处理也变得具有挑战性。一个应用可能有很多不同的视图，MVCB设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理。业务流程的处理交予模型(Model)处理。比如一个订单的视图只接受来自模型的数据并显示给用户，以及将用户界面的输入数据和请求传递给控制和模型。

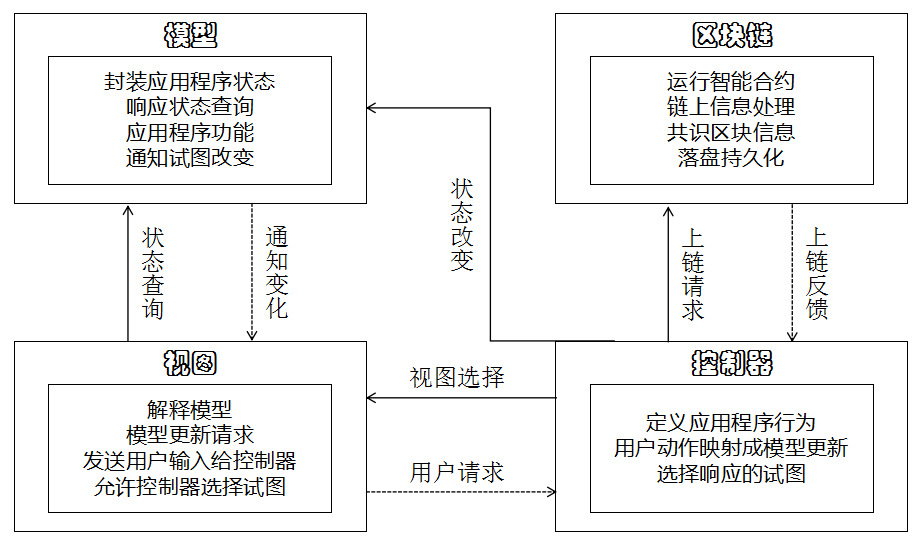
模型(Model)：就是业务流程/状态的处理以及业务规则的制定。业务流程的处理过程对其它层来说是黑箱操作，模型接受视图请求的数据，并返回最终的处理结果。业务模型的设计可以说是MVCB最主要的核心，它从应用技术实现的角度对模型做了进一步的划分，以便充分利用现有的组件，但它不能作为应用设计模型的框架。MVCB设计模式把应用的模型按一定的规则抽取出来，应该组织管理这些模型，以便于模型的重构和提高重用性。

业务模型还有一个很重要的模型那就是数据模型。数据模型主要指实体对象的数据保存（持续化）。从数据库获取数据表单信息，并将这个模型单独列出，所有有关数据库的操作只限制在该模型中。

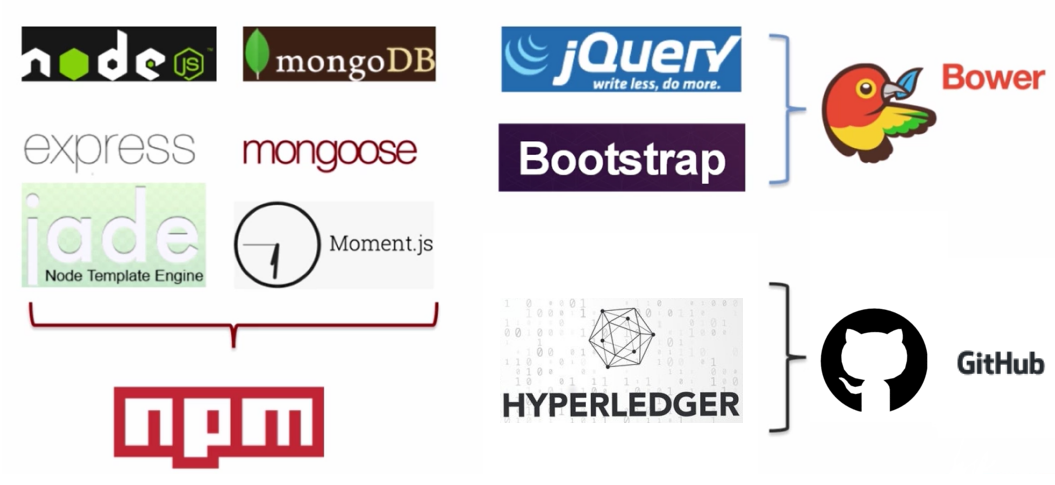
控制(Controller)：可以理解为从用户接收请求, 将模型与视图匹配在一起，共同完成用户的请求。划分控制层的作用也很明显，它清楚地告诉你，它就是一个分发器，选择什么样的模型，选择什么样的视图，可以完成什么样的用户请求。控制层并不做任何的数据处理。

区块链（Blockchain）：Blockchain service是应用程序中区块链部分，提供透明、不可篡改的分布式存储功能，并使用智能合约完成业务逻辑功能。

模型、视图与控制器的分离，使得一个模型可以具有多个显示视图。如果用户通过某个视图的控制器改变了模型的数据，所有其它依赖于这些数据的视图都应反映到这些变化。因此，无论何时发生了何种数据变化，控制器都会将变化通知所有的视图，导致显示的更新。这实际上是一种模型的变化-传播机制。模型、视图、控制器三者之间的关系和各自的主要功能，如图所示。



3.4 技术选型



后端技术：

整个后端是由Node.JS来驱动的，在这个基础之上使用express框架来快速搭建web应用。数据库采用mongoDB，以及对mongoDB快速建模的工具mongoose。后端的模板引擎使用jade。关于日期的格式化使用Moment.js。

1. Node.JS

Node.js是一个Javascript运行环境(runtime)，发布于2009年5月，由Ryan Dahl开发，实质是对Chrome V8引擎进行了封装。Node.js对一些特殊用例进行优化，提供替代的API，使得V8在非浏览器环境下运行得更好。

V8引擎执行Javascript的速度非常快，性能非常好。Node.js是一个基于Chrome JavaScript运行时建立的平台，用于方便地搭建响应速度快、易于扩展的网络应用。Node.js 使用事件驱动，非阻塞I/O 模型而得以轻量和高效，非常适合在分布式设备上运行数据密集型的实时应用。

1. express

express是一个简洁而灵活的node.js Web应用框架，提供一系列强大特性帮助你创建各种Web应用。express不对node.js已有的特性进行二次抽象，我们只是在它之上扩展了Web应用所需的功能。丰富的HTTP工具以及来自Connect框架的中间件随取随用，创建强健、友好的API变得快速又简单。

1. mongoDB

mongoDB是一个基于分布式文件存储的介于关系数据库和非关系数据库之间的数据库。由C++语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。

1. mongoose

mongoose是在node.js异步环境下对mongoDB进行便捷操作的对象模型工具。

1. Jade

Jade是一个高性能的模板引擎，它深受Haml影响，它是用JavaScript实现的，并且可以供Node使用。

1. Moment.js

Moment.js是一个JavaScript日期处理类库，用于解析、检验、操作、以及显示日期。

前端技术：

我们将使用符合规范的HTML5页面作基于区块链的脱贫攻坚应用管理平台的前端展示技术，选用jQuery类库，以及Bootstrap样式框架，它们都是网站前端的静态资源。

1. jQuery

jQuery是一套跨浏览器的JavaScript函数库，用于简化HTML与JavaScript之间的操作。全球前一万个访问最高的网站中，有65%以上的网站使用了jQuery。

使用jQuery，可以提升HTML DOM操作的性能，提高JavaScript代码的可扩展性，减少代码字节数并间接加速网页的加载时间。

1. HTML5

HTML5是用于取代1999年所制定的HTML 4.01和XHTML 1.0标准的HTML [1]（标准通用标记语言下的一个应用）标准版本。HTML 5有两大特点：首先，强化了Web网页的表现性能。其次，追加了本地数据库等Web应用的功能。广义论及HTML5时，实际指的是包括HTML、CSS和JavaScript在内的一套技术组合。

在不牺牲性能和语义结构的前提下，CSS3中提供了更多的风格和更强的效果。此外，较之以前的Web排版，Web的开放字体格式（WOFF）也提供了更高的灵活性和控制性。

1. Bootstrap

Bootstrap是一款由Twitter推出的前端框架，它的优势在于页面响应式布局、Grid布局、高可用性CSS样式表及HTML/JavaScript组件。但由于其对于IE老版本浏览器兼容性较差，因此我们将其运用在管理后台页面中，以提升后台操作的便利性及友好性。

区块链：

根据基于区块链的脱贫攻坚应用管理平台的业务模式——即以“政府主导、企业主体、市场运作、风险可控”的原则，采用联盟链的区块链技术最为合适。因此区块链底层基础平台采用Hyperledger fabric的应用框架，共识机制采用PBFT算法，保证数据的共识可信。根据不同业务灵活采用非对称加密或对称加密算法进行加密，非对称加密采用ECC椭圆曲线算法，对称加密采用AEC算法。摘要采用SHA256和SHA384哈希算法。基于PKI的成员管理，对接入的节点和客户端进行管控。