**概要设计说明书**

# 1概述

（这部分论述整个系统的设计目标，明确地说明哪些功能是系统决定实现而哪些时不准备实现的。同时，对于非功能性的需求例 如性能、可用性等，亦需提及。需求规格说明书对于这部分的内容来说是很重要的参考，看看其中明确了的功能性以及非功能性的需求。）

## 1.1系统简述

该产品提供了基于区块链技术的金融供应链解决方案。利用区块链分布式，不可篡改，不可抵赖的特性，建立由上游、核心企业、下游、银行等多方共同维护的数据平台，使平台可信。并且利用数字签名技术，所有上链数据都进行数据签名，通过智能合约固化数据操作流程，使数据可信。提供App及相关软件，记录物流及合同信息，并将信息hash记录到链上进行存证，信息本身记录到关系数据库中。为平台数据的最高级信息安全保障。

## 1.2软件设计目标

该产品的设计目标是实现用户的注册与登陆，并在登陆的时候判断该用户是代表哪一个参与方，跳转到该参与方的操作页面；实现各类合同（保险合同、运输合同等）的签名、hash计算以及上链保存，并且实现在合同上链之后的查询和更新功能；实现各种状态（如库存，运输过程）的信息上链保存，并且实现在此之后的查询与更新功能；实现token授予、支付、承兑的操作。

## 1.3参考资料

《区块链开发指南》

《FISCO-BCOS开发指南》

## 1.4修订版本记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本修订历史记录 | | | | | |
| 序号 | 修订内容 | 修订人 | 修订日期 | 版本号 | 备注 |
| 1 | 新版下发 |  | 2018-10-20 | 1.0 |  |
| 2 | 架构修改 |  | 2018-10-30 | 2.0 |  |

# 2术语表

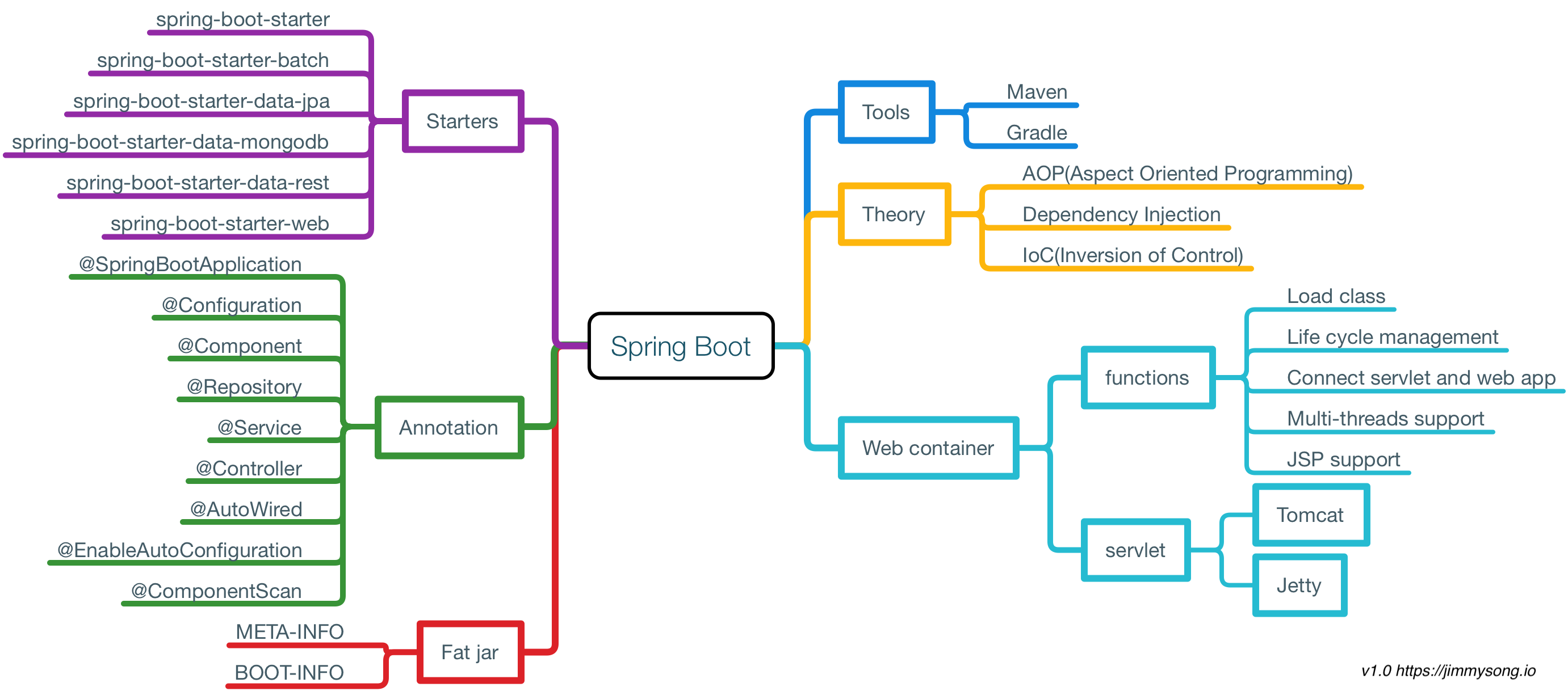
|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| 上链 | 将数据以区块的形式传输给区块链 |
| Token | 一种支付货币形式 |

# 3设计概述

## 3.1系统结构设计

**3.1.1.系统逻辑架构（软件的技术架构）**

**Spring boot架构**



**这种架构将传统的模型层被拆分为了业务层(Service)和数据访问层（DAO, Data Access Object）。** 在 Service 下可以通过 Spring 的声明式事务操作数据访问层，而在业务层上还允许我们访问 NoSQL ，这样就能够满足异军突起的 NoSQL 的使用了，它可以大大提高互联网系统的性能。

* **特点：**  
  结构松散，几乎可以在 Spring MVC 中使用各类视图  
  松耦合，各个模块分离  
  与 Spring 无缝集成

**3.1.2.系统物理架构 （物理设备架构图：软件如果部署到物理设备上）**

PC

Ubuntu

MySQL

PC

PC

客户端 Linux服务器（含链） 本地数据库

**3.1.3.系统数据模型（系统逻辑数据模型）**

## 3.2非功能设计

1. 界面美观简洁，实现UI设计的8大黄金规则。