

TTC Blockchain Contest

JAM Study

Team-10

오상문, 양승권

Nov. 2020

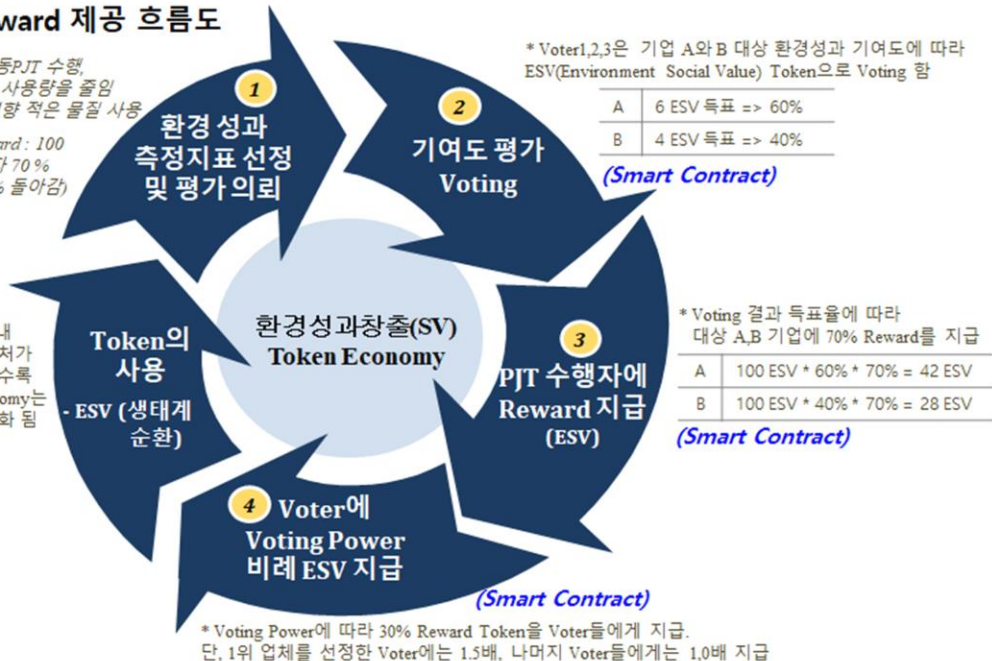


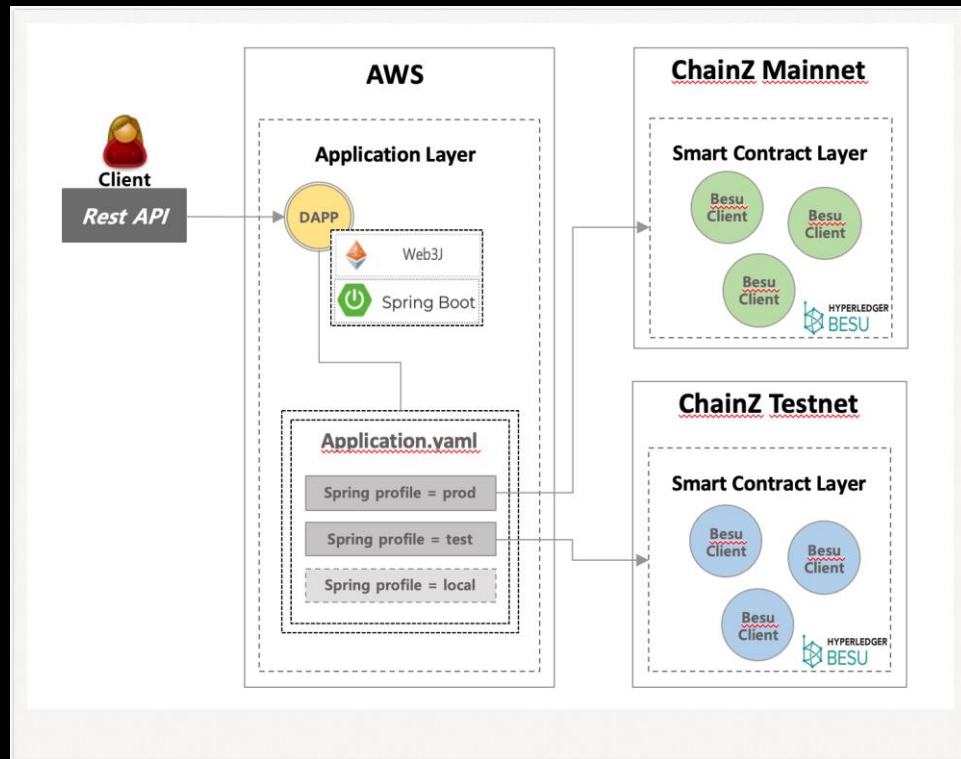
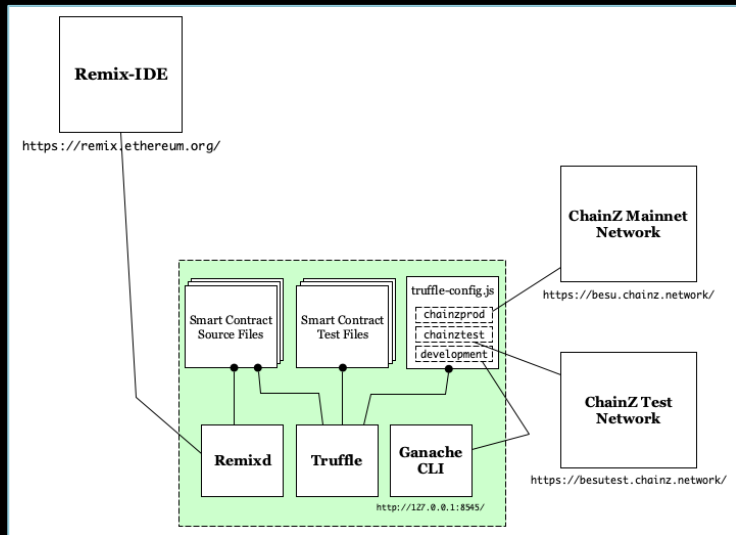
□ 환경 성과 기여 Reward 제공 흐름도

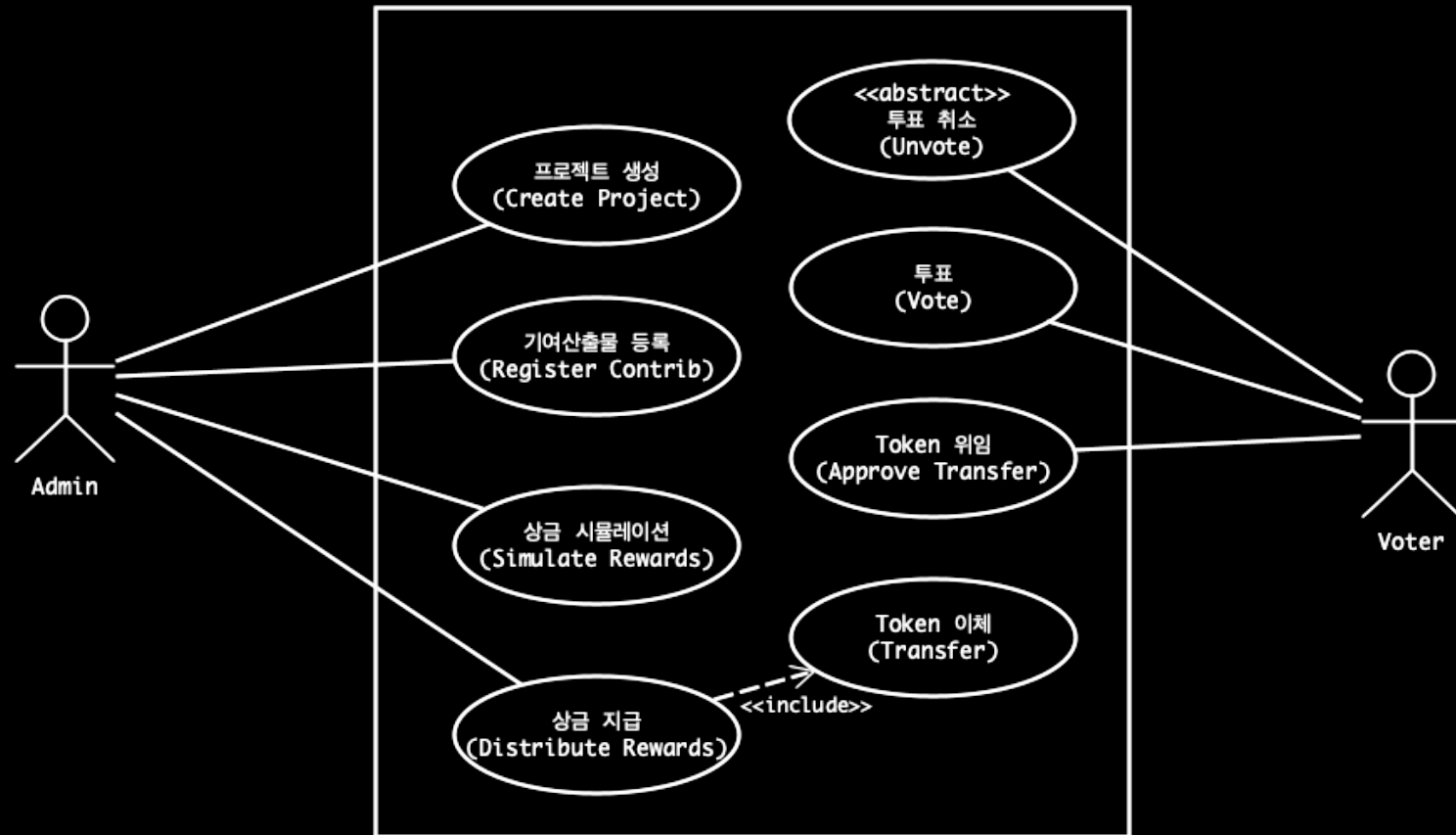
* A, B가 공동PJT 수행,
A: 화학물 사용량을 줄임
B: 환경 영향 적은 물질 사용

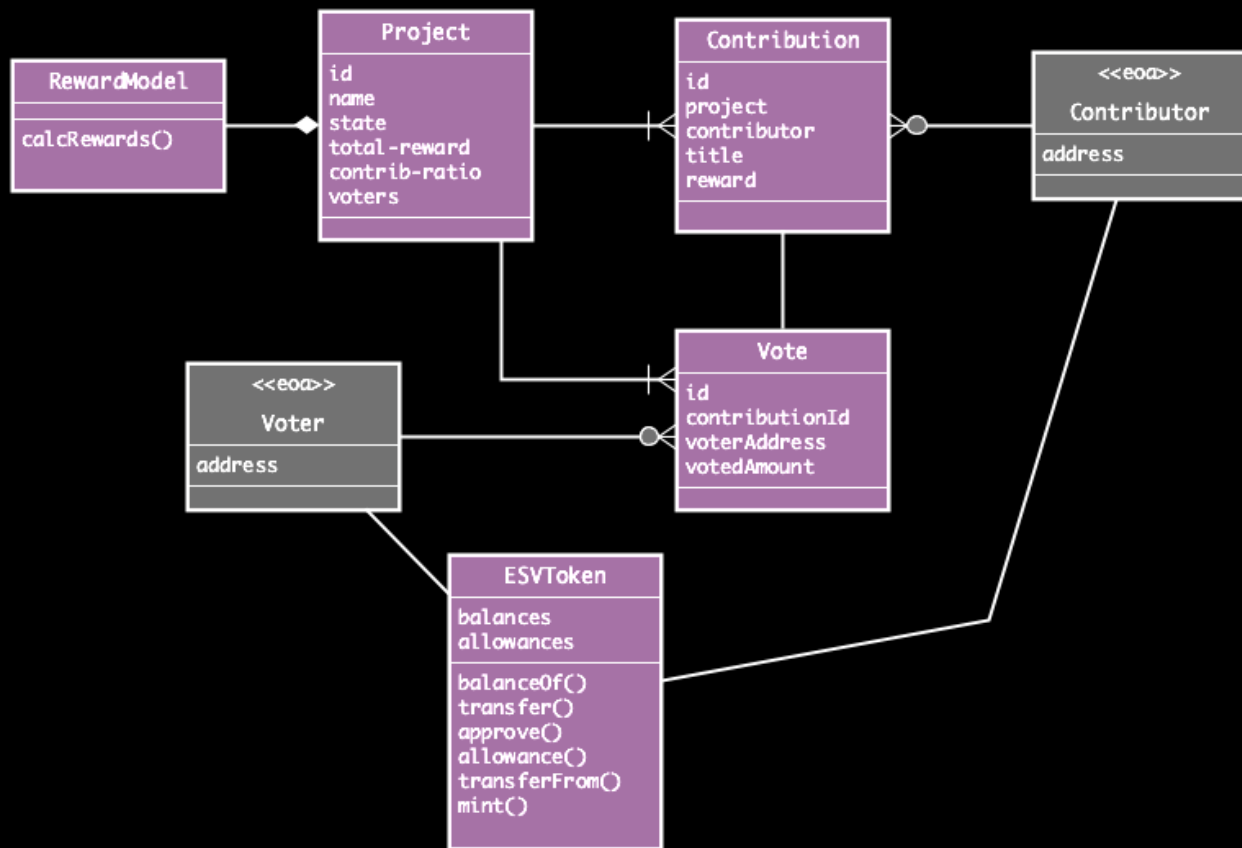
* Total Reward : 100
(PJT수행자 70 %
Voter 30 % 돌아감)

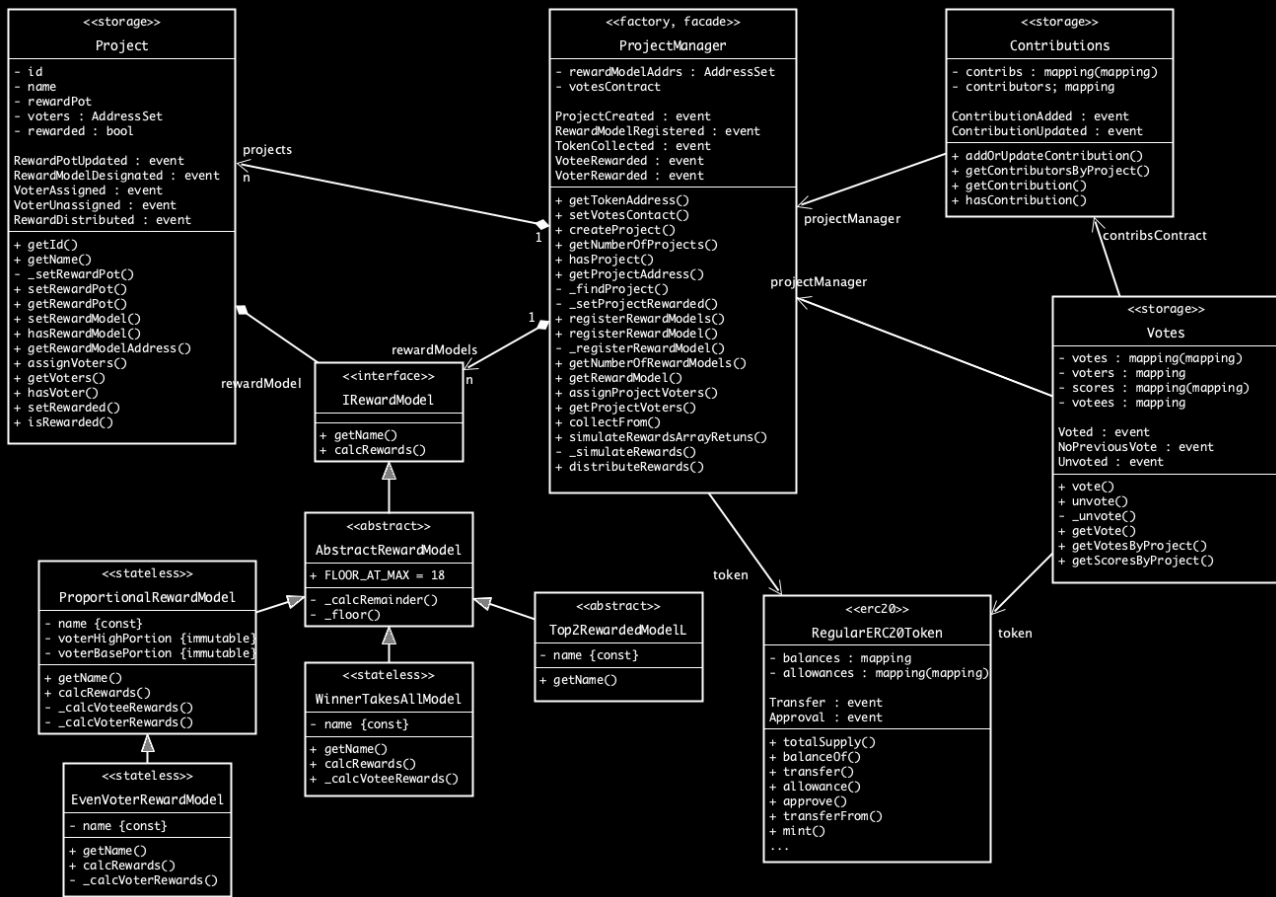
* JDP내
사용처가
많을수록
Economy는
활성화 됨

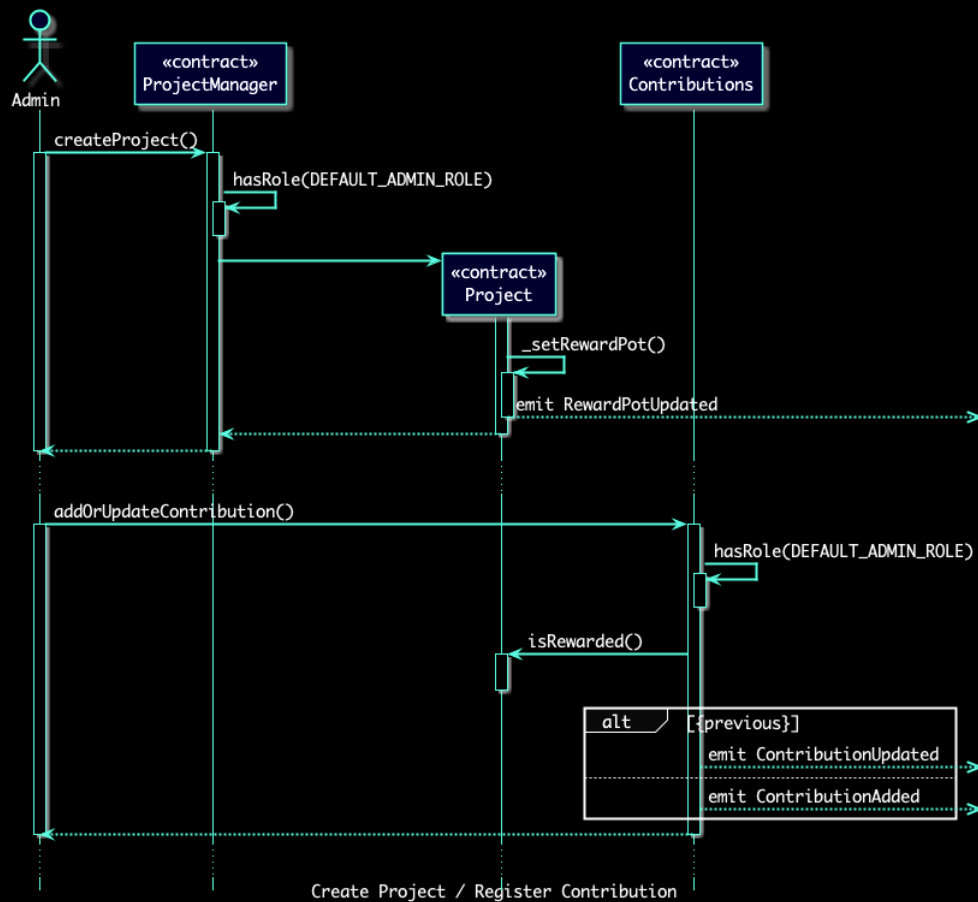


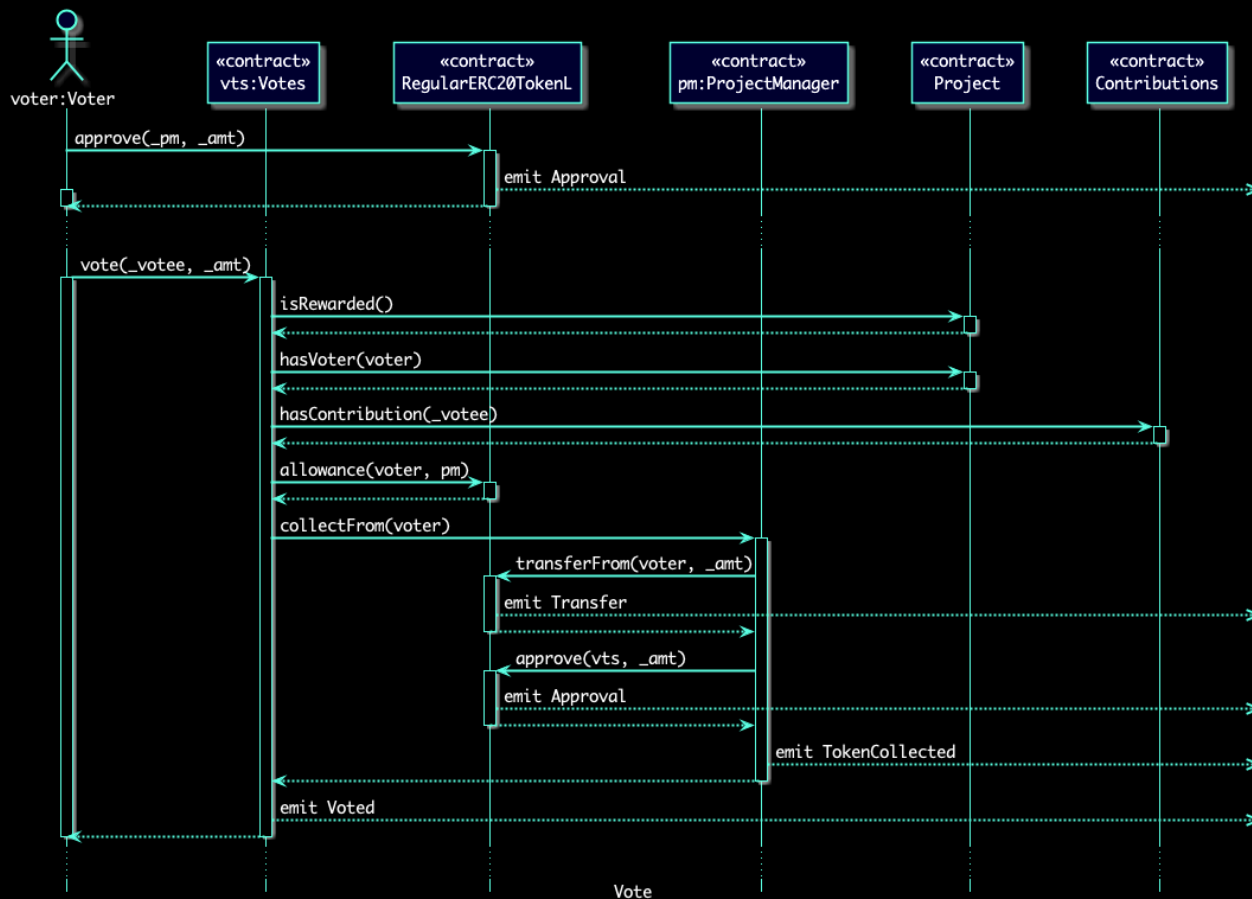


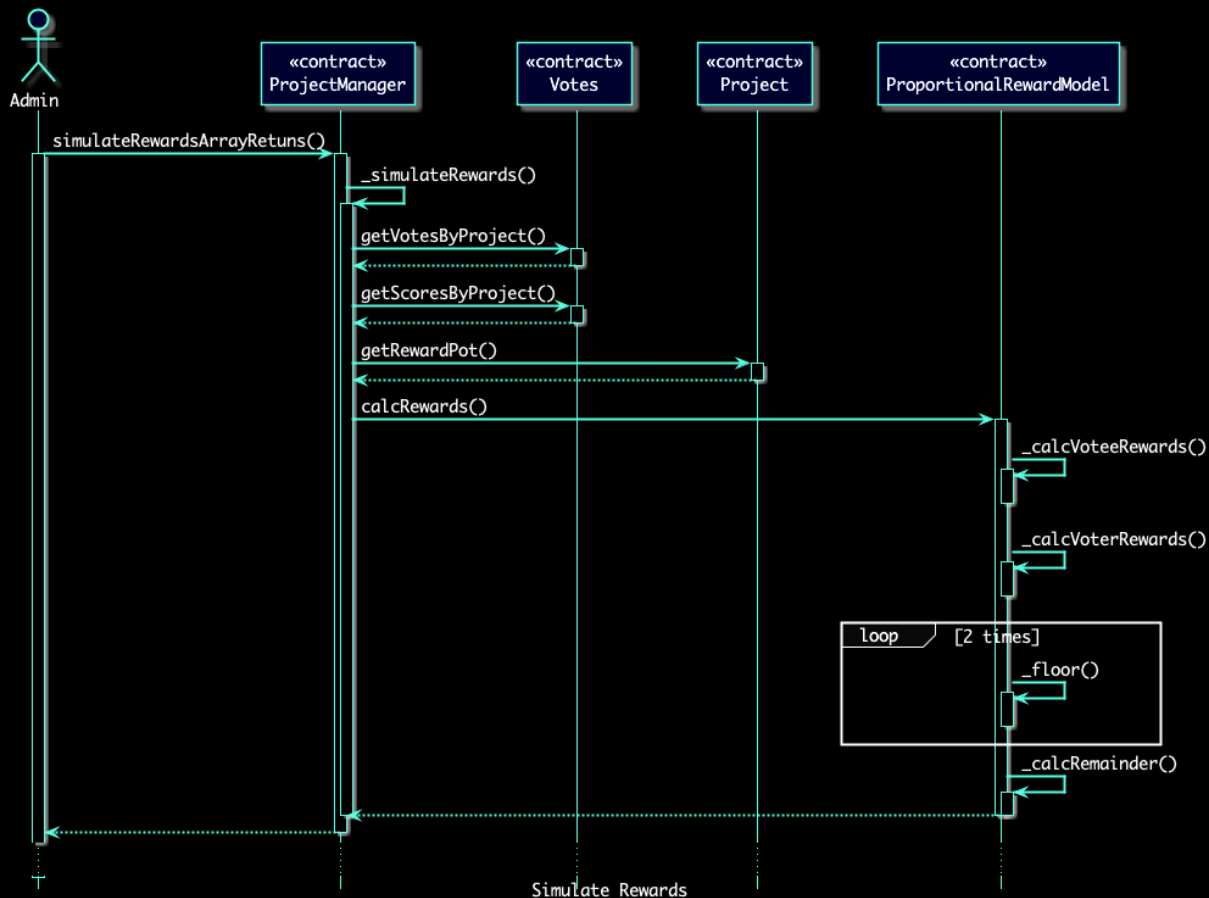


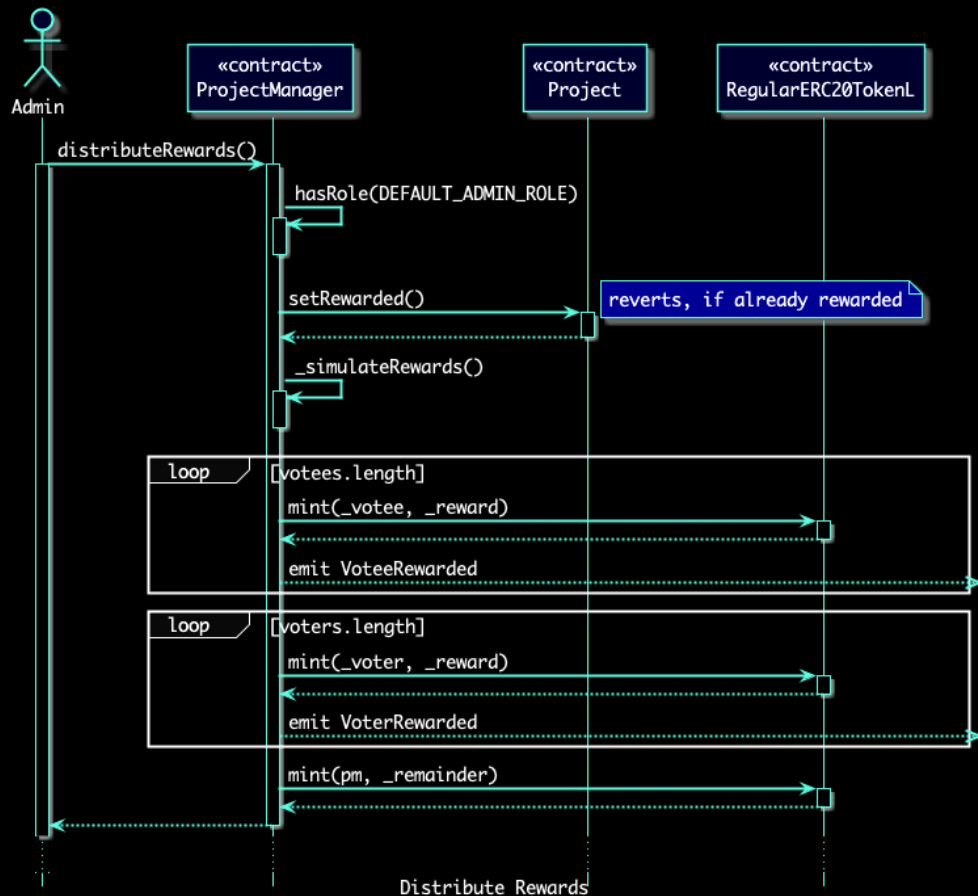












Feature	Description
<i>On-chain First</i>	<ul style="list-style-type: none"> ✓ Project, Contribution, Vote, Token 등 주요 ledger(state, 원장) 데이터와 상금 계산 logic 을 모두 smart contract 으로 구현 ✓ 일부 원장 데이터, 상금 계산 logic 등이 off-chain으로 빠지면 서비스 전체 측면에서 무결성은 일부 훼손될 수 있으므로, 주요 데이터와 logic이 가능한 on-chain 으로 구현되는 것이 매우 중요함.
<i>Modulization</i>	<ul style="list-style-type: none"> ✓ High cohesion, low coupling 기반으로 전체 logic을 다수의 contract으로 모듈화하여 설계, 구현 ✓ Software 확장성 및 재사용성, code 복잡성 해소 등을 위해서 modulization은 매우 중요한 전략
<i>Pluggable Reward Model</i>	<ul style="list-style-type: none"> ✓ 상금 (계산) 모델(model, logic) 을 순수 stateless (w/o storage) contract 으로 분리 설계 하고, solidity 가 제공하는 상속 기능을 이용하여 pluggable한 상금 모델을 추가할 수 있도록 구현 ✓ Project 마다 서로 다른 상금 모델을 mapping할 수 있으므로, 상금 logic 을 변경하거나 project 마다 다르게 적용하고자 할 때 기존 smart contract source 수정 없이 새로운 상금 모델 contract을 추가 개발하여 적용 가능 ✓ Object oriented programming에서 폭넓게 활용되는 Polymorphism(다형성) 을 smart contract 에 적용하여 확장성을 개선한 사례 ✓ <i>IRewardModel, ProportionalRewardModel, EvenVoterRewardModel, WinnerTakesALLModel</i>

Feature	Description
Expressive API using ABIEncoderV2	<ul style="list-style-type: none"> ✓ ABIEncoderV2는 다차원 배열 및 구조체 배열 등에 대한 encoding/decoding 기능을 제공하는 중요한 새기능 ✓ ABIEncoderV2를 활용하지 않을 경우, 함수 입출력 parameter type에 다차원 배열, 문자열 배열, 구조체 배열을 사용하지 못하는 심각한 제약사항이 있음. ✓ Solidity v0.6 이후부터 ABIEncoderV2를 지원 ✓ 10조 개발 프로젝트에서는 ABIEncoderV2를 적용하여 expressive 하고 활용성이 높은 함수를 정의하고 사용
Best Practices Reuse	<ul style="list-style-type: none"> ✓ OpenZeppelin contract 은 간단한 utility 성격의 library로부터 ERC20, ERC721 등 가장 많이 사용되는 EIP 표준에 대한 구현체까지 신뢰하고 재사용할 수 있는 다양한 best practice 를 제공 ✓ 검증된 best practice를 재활용함으로써, 개발 생산성 향상 및 품질 확보 등의 효과를 얻을 수 있음 <ul style="list-style-type: none"> • 'AccessControl' contract - role-based access control mechanisms • 'SafeMath' library - overflow/underflow aware(chekcing) arithmetic • 'EnumerableSet' library - enumerable(iteratable) set • 'EnumerableMap' library - enumerable(iteratable) mapping • 'ERC20PresetMinterPauser' contract - ERC20 implementation ✓ 'ERC20PresetMinterPauser' contract을 상속하여 프로젝트에서 사용한 'RegularERC20Token' contract 은 단 12줄로 구현

Feature	Description
<i>Automated Integrated Test</i>	<ul style="list-style-type: none"> ✓ Truffle test framework 을 활용하여 [프로젝트 생성 → 기여산출물 등록 → 투표 → 상금 시물레이션 → 상금 지급]까지 주요 use-case 기본 흐름을 수행하고 검증하는 통합 테스트 케이스 프로그램 작성/활용 <ul style="list-style-type: none"> • test/v2/TestIntegratedScenarios.js ✓ 경연 대회 시나리오를 포함하여 서로 다른 4개의 시나리오를 적용하여 테스트 수행 - smart contract 검증에 중요한 역할 ✓ 새로운 시나리오 또는 새로운 상금 모델 contract을 검증하려 할 때 유용하게 재사용 가능
<i>Java based REST Service</i>	<ul style="list-style-type: none"> ✓ Spring Boot, Web3j 를 활용하여 smart contract의 주요 use-case 를 외부에서 편리하게 연계할 수 있는 RESTful API 기반의 응용 application 개발