

COMP6452 Group Demo 2

Team: The Truth

Members: Anqi Zhu (z5141541),
Arunima Lahiri (z5187023),
Ziqing Yan (z5161462)

Context

- Recap Domain Background & Problem
- Finalized architecture design
- Implementation Detail
 - Blockchain platform
 - Database
 - Other techniques used for off-chain components
- Demo
- Source code walk through

Background & Domain

Crowd-sourced Review Platform

Development of a crowd-sourced review platform with blockchain technology helps to improve the reliability, security and transparency of traditional centralized review platform.

Challenges

Challenge 1: Avoid reviews of fake experience for malicious purpose

Solution: Prove that the customers' reviews are provided from their real experience, and the proof would be recorded permanently, immutably and transparently.

Functional Requirement:

1. User can post review for specific restaurant which they has been visit
2. System needs to identify the review is reliable based on the restaurant database and user's receipt number or other payment records that can prove that they have visited that specific restaurant.

Non-functional Requirement:

1. Security of use of external database
2. Consistency of restaurant information and users review information
3. Stability of post system, make sure that the post review not fail under regular environment

Challenge 2: Encourage a high-quality review sharing platform

Solution: Mimic a block-mining-like encouraging system for attentive efforts.

Functional Requirement:

1. Users can receive credits depends on how many user likes their reviews and who likes their reviews
2. Users can receive rewards such as coupons/discounts when their credit reach a specific level

Non-functional Requirement:

1. Reliability of credit earn system
2. Security of credit transfer
3. Flexibility of exchange system e.g. coupons and discounts are use under conditions set by restaurants and system

Challenge 3: Value the credit of the reviews without third-party interference but still reliable and secure

Solution: Mimic a proof-of-state mechanism to value the credit of both users and reviews via interactive credible user comments. Heavy data management might be confronted, but still want transparency.

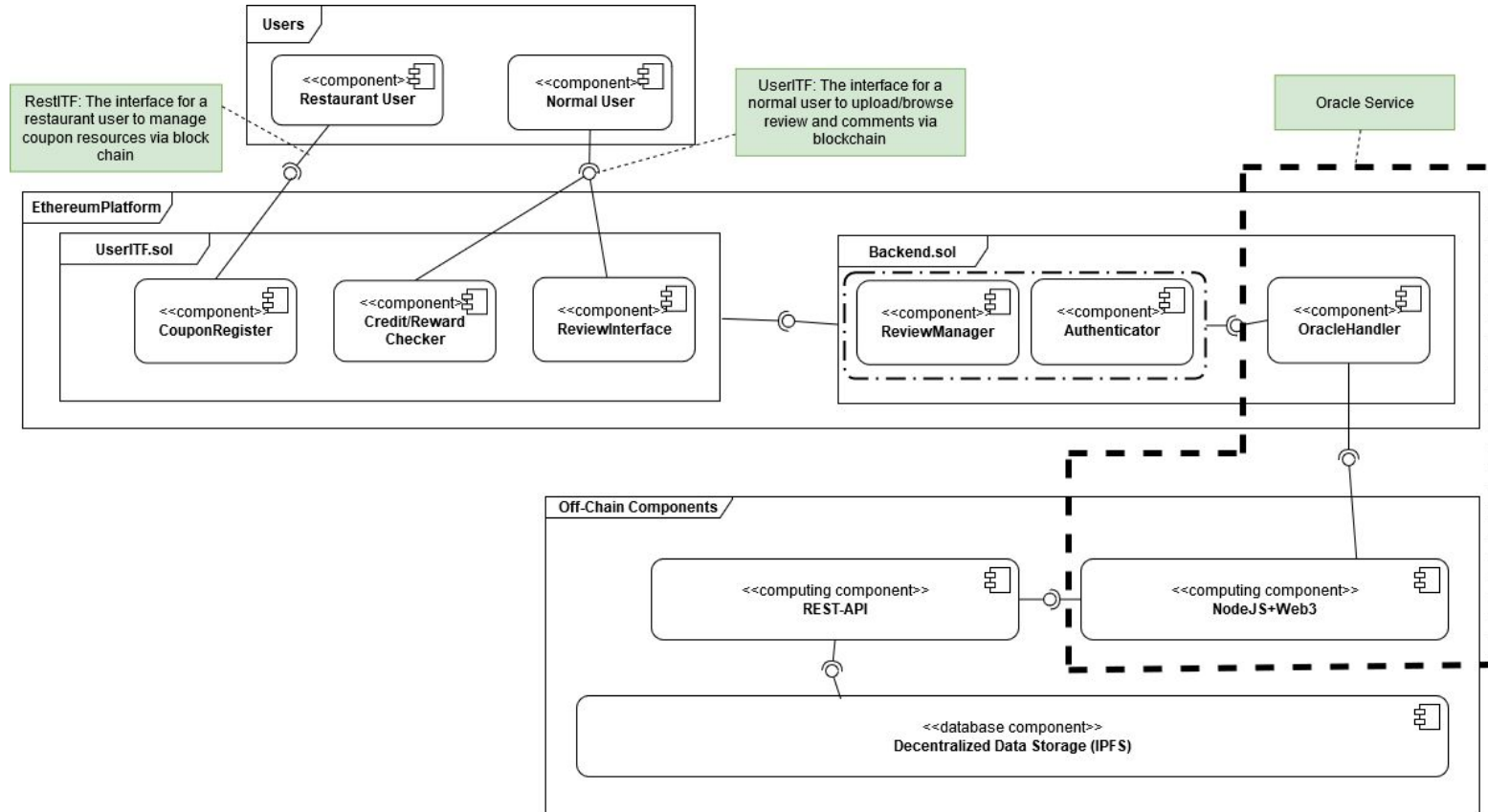
Functional Requirement:

1. The reliability of a user review is measured by quantified credits.
2. The credits of a review are piled up by the agreements/disagreements from other credible users.
3. The amount of reward from the agreement/disagreement is also measured with quantified credits by considering the commenter's credit.
4. A review should be awarded with basic credits when posted into the system. The amount of the credits depend on the credits of its writer.

Non-functional Requirement:

1. Transparency of the interactive data and records
2. Lower data storage cost
3. Security of the data storage

Architecture Design



Blockchain Component

UserITF.sol

- User Details
- Account Management Functions
- Review/Comment Management Interfaces
- Credit/Prize Management Interfaces
- Callback Functions

Backend.sol

- Review/Comment Storage Screenshots
- Receipt Authentication Interface
- Oracle(IPFS) Request Handler
- Callback Functions

StringTools.sol

- A set of handy functions mostly related to string analyzing with other types or formats

NodeJS + Web3

in a Truffle Framework

Ganache-Cli

- Initiate a server and start hearing in-net objects

Oracle.js

- Monitor and conduct off-chain interactions with the specified compiled+deployed oracle contract to provide a set of Oracle services

REST-API & IPFS

Node.js (Development of REST API)

- Products.js (records entry, manage receipts, reviews, comments)
- Coupon.js (coupons entry, manage coupons exchange and usage)

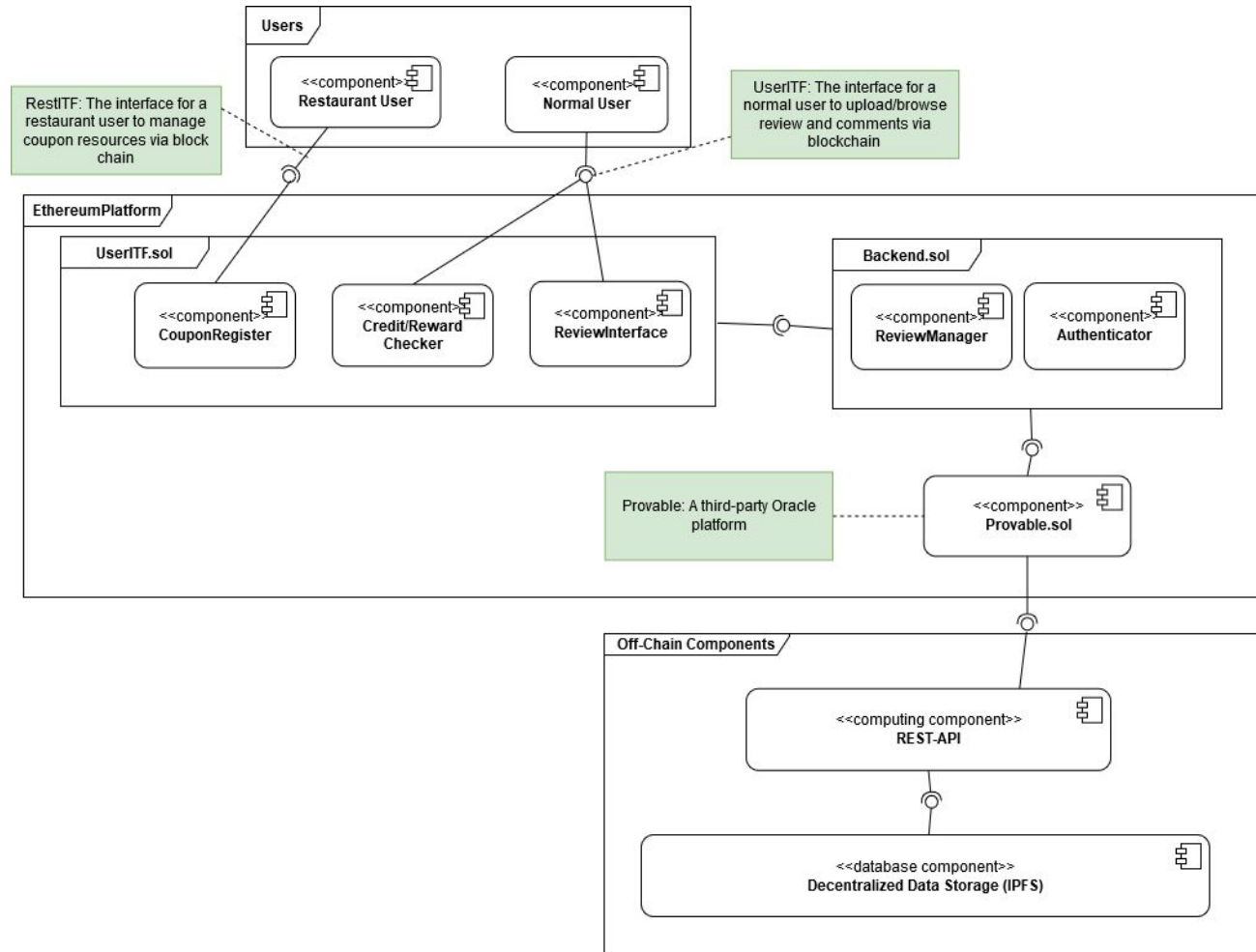
Connection of IPFS and Node.js (through IPFS API)

- Products.js (upload the reviews and comments file to IPFS and store its hash)

Published on Heroku app in public

Provable Platform

- Third party Oracle Interface
- Inherit the services by adding “usingProvable” at the end of each contract definition
- Implemented for
 - Receipt Authentication from a separated database on an API
 - Review/Comment Storage towards IPFS
 - Coupon Management with different types
 - On-chain Review Browsing functions
- The waiting time could be very long, and the connection is vulnerable



Demo

Code Walk Through