**Software Documentation**

**Understanding Surrounding Blockchains**

The runtime of a blockchain is the business logic that defines its behaviour. In Substrate-based chains, the runtime is referred to as the "state transition function"; it is where Substrate developers define the storage items that are used to represent the blockchain's state as well as the functions that allow blockchain users to make changes to this state. Each Substrate node contains a runtime. The runtime contains the business logic of the chain. It defines what transactions are valid and invalid and determines how the chain's state changes in response to transactions. The "outer node", everything other than the runtime, does not compile to Wasm, only to native. The outer node is responsible for handling peer discovery, transaction pooling, block and transaction gossiping, consensus, and answering RPC calls from the outside world. While performing these tasks, the outer node sometimes needs to query the runtime for information or provide information to the runtime. A Runtime API facilitates this kind of communication between the outer node and the runtime.

An extrinsic is a piece of information that comes from outside the chain and is included in a block. Extrinsics fall into three categories: inherents, signed transactions, and unsigned transactions.A block in Substrate is composed of a header and an array of extrinsics. The header contains a block height, parent hash, extrinsics root, state root, and digest. This section will only focus on the extrinsics root. Extrinsics are bundled together into a block as a series to be executed as each is defined in the runtime. The extrinsics root is a cryptographic digest of this series. This serves two purposes. First, it prevents any alterations to the series of extrinsics after the header has been built and distributed. Second, it provides a means of allowing light clients to succinctly verify that any given extrinsic did indeed exist in a block given only knowledge of the header.

Blockchain nodes use consensus engines to agree on the blockchain's state. A blockchain runtime is a state machine. It has some internal state, and state transition function that allows it to transition from its current state to a future state. In most runtimes there are states that have valid transitions to multiple future states, but a single transition must be selected.

Blockchains must agree on:

* Some initial state, called "genesis",
* A series of state transitions, each called a "block", and
* A final (current) state.

In decentralized systems, the nodes will see transactions in different orders, and thus they must use elaborate method to exclude transactions. As a further complication, blockchain networks strive to be fault tolerant, which means that they should continue to provide consistent data even if some participants are not following the rules. Some block construction methods are:

Aura (round robin) - Aura provides a slot-based block authoring mechanism. In Aura a known set of authorities take turns producing blocks.

BABE (slot-based) - slot assignment is based on the evaluation of a Verifiable Random Function (VRF). Each validator is assigned a weight for an epoch. This epoch is broken up into slots and the validator evaluates its VRF at each slot. For each slot that the validator's VRF output is below its weight, it is allowed to author a block.

Slot-based consensus algorithms must have a known set of validators who are permitted to produce blocks. Time is divided up into discrete slots, and during each slot only some of the validators may produce a block. The specifics of which validators can author blocks during each slot vary from engine to engine. Substrate provides Aura and Babe, both of which are slot-based block authoring engines.

A fork choice rule is an algorithm that takes a blockchain and selects the "best" chain, and thus the one that should be extended. The longest chain rule simply says that the best chain is the longest chain.

Some nodes in a blockchain network are able to produce new blocks, a process known as authoring. Exactly which nodes may author blocks depends on which consensus engine you're using.

Users in any system want to know when their transactions are finalized, and blockchain is no different. In some traditional systems, finality happens when a receipt is handed over, or papers are signed.

Using the block authoring schemes and fork choice rules described so far, transactions are never entirely finalized. There is always a chance that a longer (or heavier) chain will come along and revert your transaction. However, the more blocks are built on top of a particular block, the less likely it is to ever be reverted. In this way, block authoring along with a proper fork choice rule provides probabilistic finality.

When deterministic finality is desired, a finality gadget can be added to the blockchain's logic. Members of a fixed authority set cast finality votes, and when enough votes have been cast for a certain block, the block is deemed final. In most systems, this threshold is 2/3. Blocks that have been finalized by such a gadget cannot be reverted without external coordination such as a hard fork.

GRANDPA validators vote on chains, not blocks, i.e. they vote on a block that they consider "best" and their votes are applied transitively to all previous blocks. Once more than 2/3 of the GRANDPA authorities have voted for a particular block, it is considered final.

**Cover Page with the name of your consultancy company**

**Table of contents**

**Problem statement** **/2**

**Idea surrounding Business Blockchain:**

Creating a blockchain that links businesses, financial firms, and banks together in a private, permissioned blockchain where each user is known to each other. The transactions will have some privacy, possibly zero knowledge proofs, which allows for the blockchain to be used by numerous different businesses even if they are not interacting with each other for the processes of Source / Procure to pay and quote to cash between the businesses. Self-validating sub-ledgers for receivables and payables for the business so that administration costs are minimised. Blockchain will help supply chain partners with some of their challenges by creating a complete, transparent, tamperproof history of the information flows, inventory flows, and financial flows in transactions.

with blockchain, the retailer records the digital token for the order. The supplier then logs in the order and confirms to the retailer that the order has been received—an action that again gets recorded on the blockchain but would not generate an entry in a financial ledger. Next the supplier requests a working-capital loan from the bank to finance the production of the goods. The bank verifies the order on the shared blockchain, approves the loan, and records the loan’s digital token on the same blockchain.

functions can be automated through smart contracts, in which lines of computer code use data from the blockchain to verify when contractual obligations have been met and payments can be issued. Smart contracts can be programmed to assess the status of a transaction and automatically take actions such as releasing a payment, recording ledger entries, and flagging exceptions in need of manual intervention.

Assets and items of monetary value that are being traded could be tokenised to enable this optimisation. Asset tokenization is a term for the use of blockchain technology to represent ownership or rights to an asset as a tradable, on-chain token. Though it most commonly refers to the tokenization of financial or fungible assets, such as shares in a company or a quantity of gold, asset tokenization can hypothetically refer to the tokenization of any material or nonmaterial thing possessing monetary value: everything from a piece of art to a patent to an hour of a skilled worker’s time. Greater transparency regarding ownership and ownership history; and a reduction in administrative costs associated with the trading of these assets, including management, issuance, and transactional intermediaries. The process of tokenization creates a bridge between real-world assets and their trading, strorage and transfer in a digital world. The corresponding basis is built by using the Blockchain technology.

Blockchain is a distributed ledger technology that enables digital assets to be transacted and traded in near real time. The record it keeps is permanent and irreversible.

Business blockchains are being used to reinvent how transactions are managed. They can take time and costs out of almost any process, enabling near real-time operations. And they deliver a high degree of accuracy and control, with much less risk than many alternatives. Blockchains perform recordkeeping using automated, low-cost mechanisms. They enable asset transfer through secure, real-time methods. And they provide governance in the form of smart contracts. A smart contract makes sure each part of a transaction is validated the instant it happens, triggering the next required action, exactly when it is supposed to occur, until the process is complete.

Business blockchains are set up by a single company or a group of companies where participants are specified and known. They are designed to improve transaction processing. Public blockchains that support cryptocurrencies like Bitcoin are an entirely different thing. Finance can generate significant value from business blockchains without having anything to do with digital currencies.

Blockchains integrate different systems to get data right at the point of origination, which can eliminate downstream reconciliations. This enables straight-through processing, also known as touchless transactions. For example, a company uses blockchain to match a customer purchase order with the buyer order, and records that action on a blockchain. Now there is one source of the truth, which is visible to both parties.

Smart contracts provide the governance mechanism for business blockchains. Once a smart contract is locked down, the terms and conditions can’t be changed unless all those affected agree.

The hard part is establishing a sustainable group of trading partners, with transactions governed by effective smart contracts and clear rules of engagement. The permanent and irreversible nature of blockchains greatly reduces the possibility of fraud and errors. Blockchains enable the management of things like asset purchases, financing, warranties, insurance, regulatory compliance, and public safety—in an integrated manner and all at the same time.

You could get instant visibility on the status of accounts receivable, supply chain movements, and other transactions. You would have a transparent, chronological history of events for a single source of truth. Blockchain can eliminate the lag in payment cycles and asset transfer, which can help reduce cost, improve accuracy, and provide compliance efficiency. Additionally, the transparency of blockchain can help streamline trade finance or supply chain financing in a multi-party network setting.







I wished to create a stablecoin in which I would utilise, so business practices did not have to rely on the absence of fluctuations of token value. However, to make a collateralised stablecoin I would need to own the fiat or assets in which the token was based upon, as ‘collateral.’ For the purpose of this conceptual design, I will utilise tokens that are not truly backed by US Dollar or another asset but in reality, this would be the case.

**Issues relevant to program /3**

Evaluate ONE social or ethical consideration in developing this program.



**Interface design /3**

Using a design tool develop a mock interface for your program's main interface. Your team needs to ensure that you consider the needs of the intended audience and address any ergonomic and design issues.



**Quality assurance criteria /2**

Describe the criteria the program needs to meet.

****

**Feasibility Study /10**

Conduct a feasibility study on the on the feasibility of your project, the report must contain the following sections:

* **Define the problem:** you can copy and paste this from your problem definition statement
* **Economic feasibility:** assess the economic feasibility of the program
* **Technical feasibility:** assess whether the program can be technically created.
* **Operational feasibility:** assess whether you can operationally design, create and maintain the program
* **Scheduling feasibility:** assess whether there will be any scheduling issues in creating the program
* **Recommendation:** recommend whether your team can develop the program

****

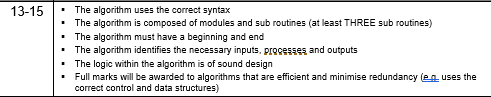
**Gantt Chart /5**

Construct a Gantt chart that outlines the tasks that need to be completed in order to design the program.



**Algorithm /15**

Using Pseudocode develop an algorithm that demonstrates the logic of proposed application.

****

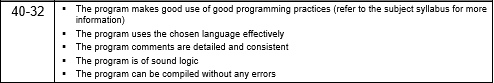
These validators are often seen to be the most wealthy, and, as a result, influence the PoS network as they are the most staked. Usually, the number of candidates to maintain the network with the necessary knowledge (and equipment) is limited; this can directly increase operational costs as well. Systems with a large number of validators tend to form pools to decrease the variance of their revenue and profit from economies of scale. These pools are often off-chain.

A way to alleviate this is to implement pool formation on-chain and allow token holders to vote [with their stake] for validators to represent them.

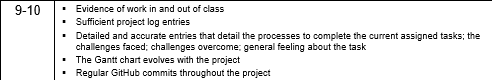
Polkadot uses NPoS (Nominated Proof-of-Stake) as its mechanism for selecting the validator set. It is designed with the roles of validators and nominators, to maximize chain security. Actors who are interested in maintaining the network can run a validator node.

There are two protocols we use when we talk about the consensus protocol of Polkadot, GRANDPA and BABE (Blind Assignment for Blockchain Extension). We talk about both of these because Polkadot uses what is known as hybrid consensus. Hybrid consensus splits up the finality gadget from the block production mechanism.

**Commented Code**

****

**Project Work Evidence**

****

**Showcase Video**

****