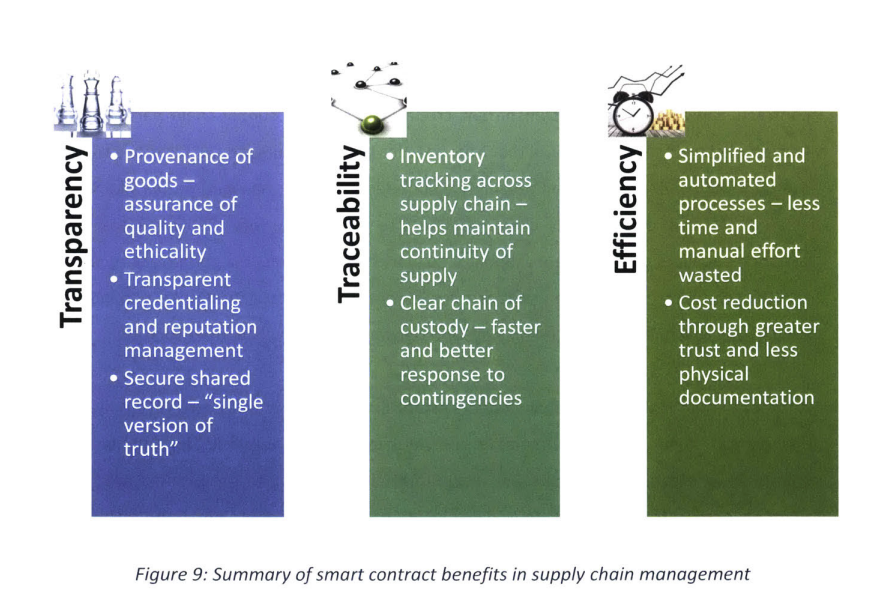
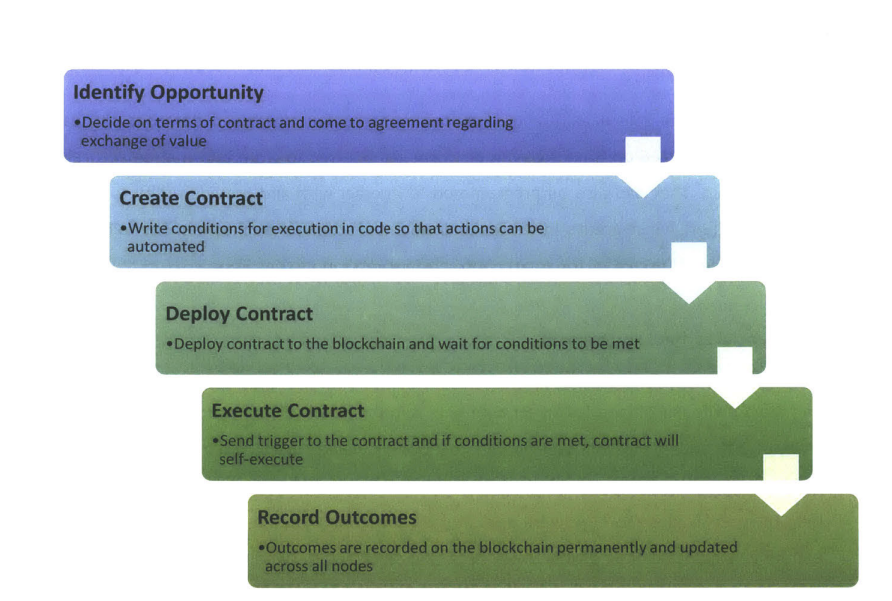
SERA Smart contract Factory

A supply chain encompasses all of the activities that go into the delivery of goods or services, beginning at the earliest stage of creation and ending at the final stage of destruction or extinction. With the impact of globalization, supply chains typically cross the boundaries of both organizations and countries. They can also vary significantly in terms of length (i.e. the number of tiers across the supply chain) and depth (i.e. the number of suppliers or customers within each tier), depending on the good or service in question. The complexity of a generic s

Current supply chain processes are long and complex. Smart contracts can help to simplify this and improve visibility across the supply chain. By coupling them with loT devices that can track the location of goods, smart contracts can allow for tracking of inventory and the chain of custody throughout a supply chain. With this information, companies will be better equipped to deal with disruptions or manage incidents such as recalls. This also reduces the risk of theft or fraud along the supply chain. On top of tracking the journey of goods, smart contracts also allow companies and consumers to determine their provenance. This is especially important in the food industry (for health reasons), but also as environmental sustainability becomes a bigger concern. Companies like Walmart, Maersk, BHP Billiton, and Everledger have started exploring the use of smart contracts to track a diverse range of goods, including meat, shipping containers, mining samples, and even diamonds





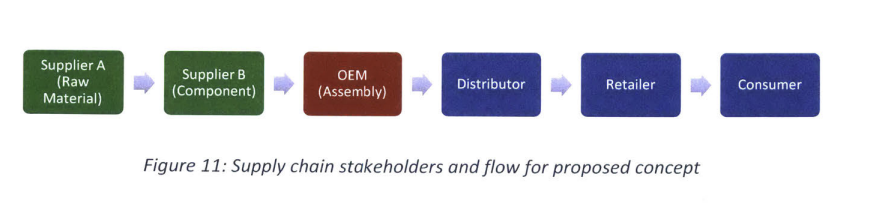
A concept is proposed that seeks to use smart contracts to address three main challenges in supply chain management:

1) Determining the provenance of goods

2) Tracking the progress of goods through the supply chain

3) Building trust through an open database of supply chain partners, including their reputation

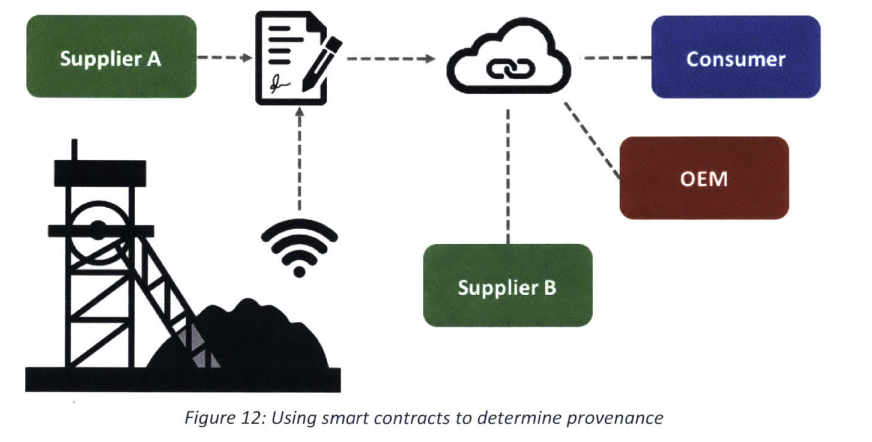
To help illustrate the concept, a simplified, generic supply chain for a basic consumer electronic product is used. The supply chain flow is shown in Figure 11. The OEM (red) is the focal organization and is in charge of coordinating the supply chain. The other parties are designated as either supply side (green) or demand side (blue). Supplier A sources the raw material and ships it to supplier B, who manufactures the individual components. The parts are then shipped to the OEM, who assembles the final product. The product is shipped to the distributor, who distributes it to the retailer to sell to the consumer



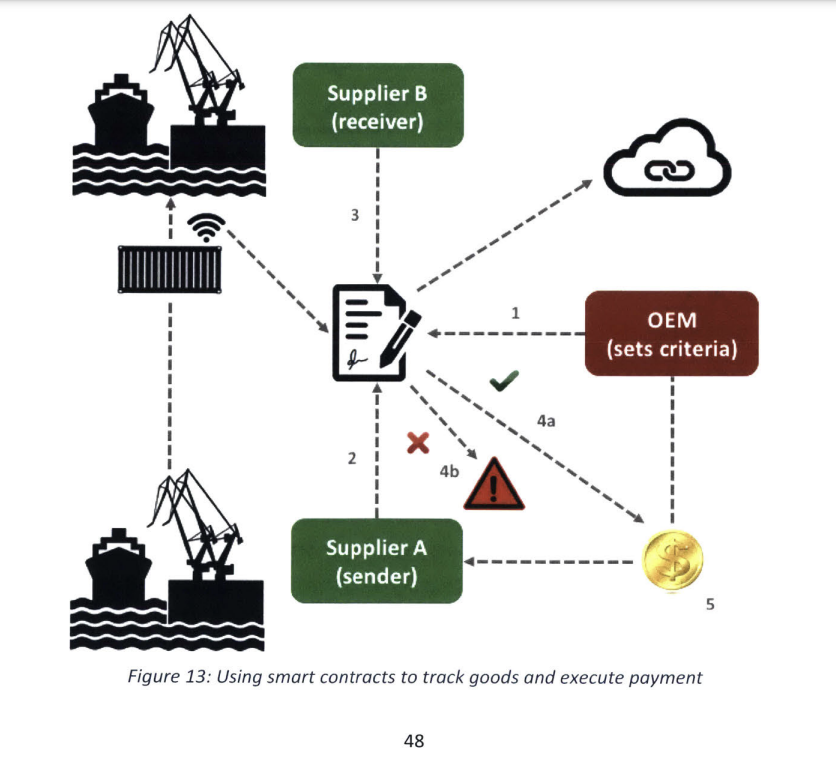
The proposed concept will run on a blockchain that all supply chain partners will be able to access (consumers can access certain functions such as checking provenance). It can be deployed and administered by either the focal organization or a neutral third party

A caveat is that some level of existing trust in the party administering the blockchain is still required, so it is not completely trustless. However, this will allow more control and mediating actions can be easily carried out in the event something goes awry.

To determine provenance, supplier A - the origin of the supply chain - will be required to record its details and the details of its raw material to the blockchain. Through integration with a smart sensor, the location and time from which the raw material is shipped can be immutably stored. This will allow any of the other parties, including the consumer, to access the information through the blockchain (see Figure 12). By providing a quick way to determine provenance, raw material providers are kept accountable and parties can easily verify the source of their product.



For tracking of the product along the supply chain, each party will be required to record the details to the blockchain whenever they send out a shipment or receive a shipment. When a shipment is received, the receiving party will confirm that everything is in order. If the shipment arrives on time and at the correct location (verified by smart sensors), a payment in the form of 47 cryptotokens is triggered to the shipping party. Figure 13 illustrates an example: the OEM predetermines the acceptable lead time and correct location for each leg of shipment (step 1). It also holds all the cryptotokens initially. Supplier A records the details on the blockchain when it ships raw material to supplier B (step 2). When supplier B receives the shipment, it records the details as well (step 3). The smart contract checks that the shipment tallies and whether the predetermined criteria have been met. If everything is in order (step 4a), a payment is executed from the OEM to supplier A (step 5); otherwise, an alert is triggered so that parties can rectify any problems (step 4b). Automated tracking and payment will help to simplify the processes within the supply chain and provide real-time updates on product status, so that parties can remain agile in the event of unforeseen circumstances.

****

1) addProducer - this allows producers/suppliers to input their details and record them to the blockchain. Solidity uses a mapping to map the producer's Ethereum public address to a struct containing its details such as name, phone number, city, state and country of origin, as well as certification. Hence, the details can be easily retrieved subsequently with the producer's public address.

2) removeProducer - this allows the administrator of the contract (either focal organization or neutral third party) to remove a producer/supplier from the database in the event of any changes. Administrator access is required to prevent unnecessary tampering

3) findProducer - this allows any party to display the details of a producer/supplier by entering their Ethereum public address. Consumers can use this to verify that the producer supplying their product is legitimate and certified. This function is costless as it does not require making any changes to the blockchain. 50

4) certifyProducer - this allows the administrator of the contract to certify a particular producer/supplier (perhaps upon receipt of necessary documentation). The certification status will be recorded on the blockchain and displayed together with the producer's other details. Administrator access is required to provide an additional layer of verification.

5) addProduct - this allows a producer/supplier to record to the blockchain each time their product is tagged at its origin. If integrated with a smart sensor, it can automatically record the location of the product. Once recorded to the blockchain, the producer's public address and block timestamp will also be automatically recorded. A mapping is used to link the product's serial number/tag with its details.

6) removeProduct - this allows the administrator of the contract to remove a product from the database in the event of any changes. Administrator access is required to prevent unnecessary tampering.

7) findProduct - this allows any party to display the details of a product by entering its serial number. It returns the producer's public address as well as location, date and time of origin. This function is costless as it does not require making any changes to the blockchain. Consumers can use this function together with the findProducer function to determine the provenance of their products.

The Tracking smart contract allows parties in a supply chain to track the shipment of goods and automatically execute payment in the form of tokens once every leg of shipment is completed, provided that certain predetermined criteria are met. There are two components to the contract - one for managing the tokens and the other for managing the shipments. Events are also used to display messages and details when transactions are executed on the blockchain. This provides more information than the default transaction hash that is displayed. The smart contract consists of the following functions (see Appendix B for the complete smart contract code): 51

1) sendToken - this allows tokens to be sent from one Ethereum account to another, by specifying the respective public addresses and the token amount. It has logic built in to check if there are enough tokens in the sender's account and to automatically update the balances of both accounts once the transaction has succeeded. Event messages are published to the blockchain to alert parties of the transactions. ( lets discuss this function in details )

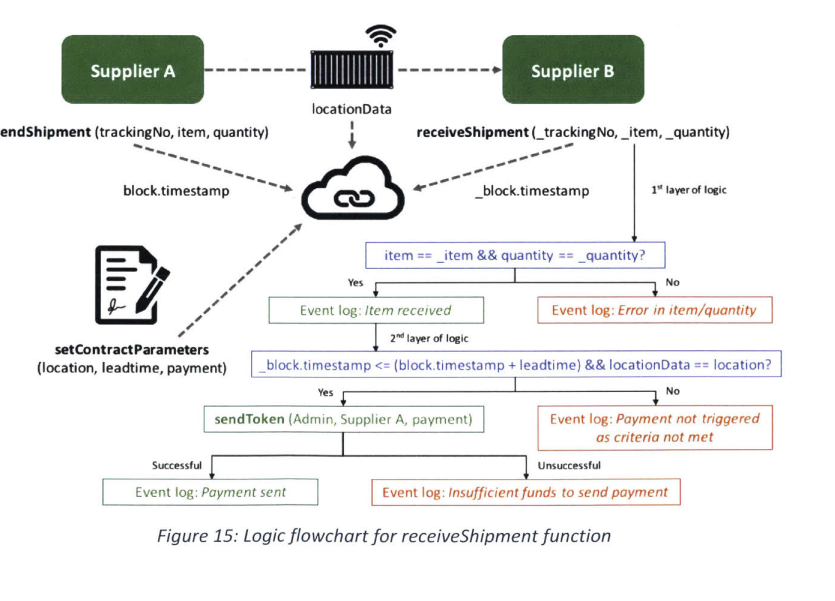
2) getBalance - this allows any party to check the token balance of an account by entering the Ethereum public address. This function is costless as it does not require making any changes to the blockchain. Upon deployment of the contract to the blockchain, the number of initial tokens can be set and all tokens are initially held by the administrator. Separate arrangements will have to be made to agree on the value of the tokens and the allocation of initial tokens.

3) recoverToken - this allows the administrator of the contract to recover tokens from any account. It is a form of check and balance to prevent parties from abusing the sendToken function (since it can be used to send tokens from any account to another account). Similar to the sendToken function, logic is built in to check for sufficient balance and automatically update balances.

4) setContractParameters - this allows the administrator of the contract to predetermine the conditions that have to be met (and have been agreed upon by parties) before a shipment is successful and payment is released. It includes details for the shipping lead time, the shipping destination, and the payment amount (in the form of tokens).

5) sendShipment - this allows the sender to record details of a shipment on the blockchain once it has been dispatched. A mapping is used to map the tracking number/tag of the shipment to details such as the item and quantity. A smart sensor can be integrated to provide real-time location data. Once recorded to the blockchain, the time of dispatch 52 and sender's address are also captured. An event is triggered to inform parties that an item has been shipped.

6) receiveShipment - this allows the receiver to record details of a shipment on the blockchain once it has arrived. Two layers of logic are incorporated into this function (see Figure 15). The first checks that the item and quantity received match the item and quantity shipped. If they match, an event is triggered to log that the item has been received successfully. Then, the next layer of logic checks to see if the predetermined shipping lead time and destination have been adhered to. If they have, the sendToken function is automatically triggered to send payment to the shipping party. Otherwise, events are logged to capture the errors (i.e. payment not sent or item/quantity do not match).



7) deleteShipment - this allows the administrator of the contract to remove a shipment from the database in the event of any changes. Administrator access is required to prevent unnecessary tampering.

8) checkShipment - this allows any party to display the details of a shipment by entering their tracking number. It returns the item, quantity, location (real-time if integrated with smart sensor), shipping timestamp, and sender. This function is costless as it does not require making any changes to the blockchain (caveat: it will not be costless if real-time location data is used).

9) checkSuccess - this allows any party to check the number of successful shipments made by a party in the supply chain, as well as their total number of shipments made. Successful shipments are defined as those that have met the predetermined criteria set in the contract (i.e. not late and delivered to the correct location). This function is costless as it does not require making any changes to the blockchain. 10) calculateReputation - this allows any party to calculate the reputation score of a supply chain partner. The reputation score is determined by taking the number of successful shipments as a percentage of the total number of shipments made by a particular party, and is expressed as an integer between 0 and 100. If a party has not made any shipments, its default reputation score is 0. This function is costless as it does not require making any changes to the blockchain. It is also essential to the proper functioning of the Reputation smart contract.

The Reputation smart contract maintains an open database of suppliers/parties in a supply chain that can be accessed by all. In addition to supplier details, it also tracks the reputation of each party. This is done by calling the Tracking smart contract that has been deployed on the blockchain and accessing the reputation score calculated in there. To make it more convenient to browse the list of suppliers, functions to filter by type of goods and reputation have been

1) addSupplier - this allows suppliers/parties to input their details and record them to the blockchain. A mapping is used to map the supplier's Ethereum public address to a struct containing its details such as name, phone number, city, state and country of origin, type of goods it specializes in, as well as reputation. Hence, the details can be easily retrieved subsequently with the supplier's public address. The address is also pushed into an array containing all of the parties' addresses.

2) removeSupplier - this allows the administrator of the contract to remove a supplier/party from the database in the event of any changes. Its public address is also removed from the array of consolidated addresses. Administrator access is required to prevent unnecessary tampering.

3) findSupplier - this allows any party to display the details of a supplier/party by entering their Ethereum public address. Parties can use this to browse suitable and preferred suppliers since information such as the type of goods and reputation are available. This function is costless as it does not require making any changes to the blockchain.

4) allSuppliers - this allows any party to display the complete list of suppliers/parties recorded on the blockchain. The array containing the list of Ethereum public addresses is returned, which parties can use to get further details. This function is costless as it does not require making any changes to the blockchain.

5) filterByGoodsType - this allows any party to search for suppliers by the type of goods that they specialize in. An array is created in memory with the same length as the complete supplier array. Logic is used to iterate across the complete supplier array and search for those with goods type matching the one specified in the query. The matches are then returned in the new array. This function is costless as it does not require making any changes to the blockchain.

6) filterByReputation - similar to the previous function, this allows any party to search for suppliers by their reputation score. An array is created in memory with the same length as the complete supplier array. Logic is used to iterate across the complete supplier array and search for those with a reputation score equal to or higher than the score specified in the query. The matches are then returned in the new array. This function is costless as it does not require making any changes to the blockchain.

7) checkReputation - this allows any party to display the reputation score of a specified supplier/party by entering their Ethereum public address. It works by calling the deployed Tracking contract and running the calculateReputation function. Since both functions do not require making any changes to the blockchain, they are costless.

8) updateReputations - this allows the administrator of the contract to derive the latest updated reputation scores of all parties and update them on the blockchain. Since the reputation scores are stored when parties add their details through the addSupplier function, they may become outdated over time as more shipments are made. Hence, the administrator can use this function to update the reputations periodically. It iterates across the complete supplier array and for each public address, updates the reputation score using the calculateReputation function from the Tracking contract. Administrator access is assigned to prevent unnecessary usage.

