# Escrow Smart-contract application requisites

Escrow smart-contracts have the purpose of working alongside PBAs, or Project Bank Accounts implementations. A PBA is a ring-fenced bank account from which payments are made directly and simultaneously to a lead contractor and members of the hierarchical contracting supply chain (Cabinet Office 2012a). The PBA may either be held in the sole name of the main contractor as trustee (‘single authority’ approach), or in the joint names of main contractor and client (‘dual authority’ approach). Under a PBA, the main contractor submits its progress payment to the client under the main contract showing a breakdown of payments to each of the suppliers. Once approved, the client pays the total amount of the progress payment into the PBA, and payment is then made from the PBA to each of the suppliers with the dual agreement of the client and main contractor. Direct payment to the suppliers from a PBA enables the traditional lengthy contractual payment credit terms, which typically exist in subcontracts within the construction industry, to be bypassed ensuring a much quicker flow of funds down through the supply chain (*Griffiths, lord - Project bank accounts the second wave of security of payment*).

Smart contracts can embed funds into a contract which will protect contractors, subcontractors and other supply chain members from insolvency and could automate the (currently manually administered) principles of payment under a PBA, increasing efficiency, decreasing payout time, and minimizing risk of fraud, back-office costs and operational risks. The use case concerning PBAs considers the use of smart contracts to automate payments within a publicly funded construction project (*Li, jenny - Blockchain in the built environment and construction industry).*

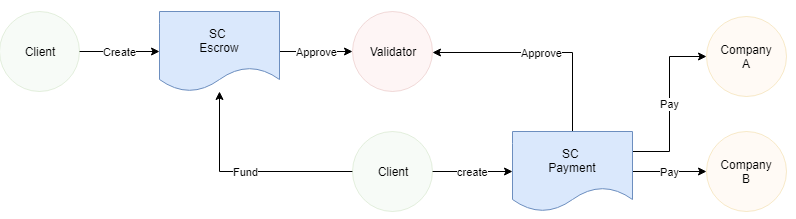
## Escrow smart-contracts Workflow

The two main advantages of using a smart-contract application, as a Project Bank Account, are the following:

1. Automate payments to companies (suppliers, contractors, sub-contractors, etc);
2. Automatic calculate pay rates and pay dates, upon agreed milestones completion.

By distributing the payments capabilities to a funded smart-contract, it is possible to make the entire process considerably more transparent, quicker and easy to check by all involved parties.

Our proposed solution follows below.



In the above example, the client (owner of the contract and the transaction executor) will, in a first instance, create the initial escrow smart-contract. This contract will hold the project, workpackage, client and company information. When it’s approved by the third-party validator, an entity responsible to validate disputes and payments, the client will create the payments smart-contract. This application will distribute funds from the escrow contract, which are sent to the payment smart-contract upon the completion of a certain milestone. Payments can be done to different companies, as there may be multiple suppliers for each escrow and payment smart-contracts.

The goal of separating both contracts is to make the process easier to audit and more transparent. On each stage disputes can happen and it becomes easier for the validator to understand where the problem may lie, as well as, other third-parties that may need access to the smart-contract data, could be granted too much information without necessity.

## Nodes and Connectors

Nodes represent the agents interacting with the smart-contracts. Agents can be (a) Clients, as in the addresses (owners) responsible for creating the smart-contracts; (b) Validators, or the third-parties responsible for validating the smart-contract code, for holding the funds and for resolving any disputes between Clients and Companies; or (c) Companies as in the agents that participate in the receiving the payments from completing work.

### Client (owner)



In this example, the Client is responsible for putting together the initial escrow. Because there is only one address owning the contract, that could be troublesome as it could be easier to attack the network. However, the more owner accounts and approval processes, the longer it takes for a decision to a consensus. In essence:

- The more owner accounts, the less risk of centralisation;

- The less owner accounts, the faster the decision-making process.

### Validator (third-party)



A Validator is an account which approves/rejects transactions from the Client, into the escrow. The idea of having a Validator, is again to mitigate risk. Much like in PBA, when we add an extra layer of entropy (complexity) to the system, we make it slower but safer. Because there is a third-party company looking for the best interests of both the Client and the Companies, in a transparent and auditable manner, all agents can agree on a given state.

- The more Validator accounts, the less risk of centralisation;

- The less Validator accounts, the faster the decision-making process.

### Company



Company nodes represent companies that are performing the work related to the project and workpackage detailed in the escrow. Disccounts and other perks can be given to companies depending on their performance. Milestones can be added in order to make payments more fluid and less dependent on approvals.

### Connectors

Each node connecter represents an action that can be taken by the node. For instance, create functionality will only be available to clients, as they’re the ones responsible for maintaining the smart-contracts, whereas companies simply update the state of each contract.

The two actions that update the state of smart-contracts are:

1. Create (update)
2. Approve

## Smart-Contracts

As explained above, smart-contracts will authenticate and validate transactions on its network, in real-time with full traceability of who does what and when. In addition to reducing contract execution related disputes, which is very common in the construction industry, this would also reduce costs associated with administration of procurement and instantly generate electronic documents in contrast to the traditional process which includes the use of hard copies of documentation and authentication by third party.

The ability to create, validate, authenticate and audit contracts and agreements in real-time, across borders, without third-party intervention, makes Blockchain and smart-contract technology appealing.

#### Escrow SC



The Escrow smart-contract is responsible for holding the information about the variables relating to the job, for which the payment is being made. Much like an invoice, escrows can hold key data about the validator company, the client, project and workpackages, as well as, the timestamp data to verify when the work was done.



The goal of the contract is to manage the entire escrow process. Other variables like location, person who completed the job, or job satisfaction state can be added as well, to improve the audit transparency.

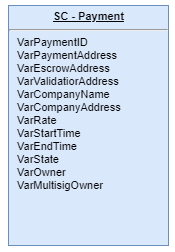
Dependencies: None

#### Payment SC



The Payment smart-contract is responsible for holding the information about the payment variables. The Payment contract will decide when the money should be paid, and to whom, based on the escrow logic. It will wait for validation from the third-party and after the timestamp is approved, payment is released to the involved parties in an autonomous way.

The important variables are the identifiers to whom the payment was made, where it came from and to what escrow was it related to.



Payments can be withheld for different reasons, like the job not being complete or problems arising. The task of the validator is to step in, when there are disagreements, otherwise the flow should be left untouched.

Dependencies: Escrow SC, Validator approval, Client approval.