www.chainfrog.com

chainfrog

Component Servicing Blockchain Use Case

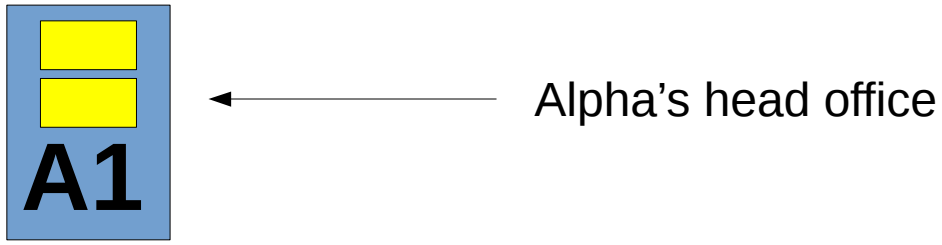
Find us at www.chainfrog.com

Email us at info@chainfrog.com

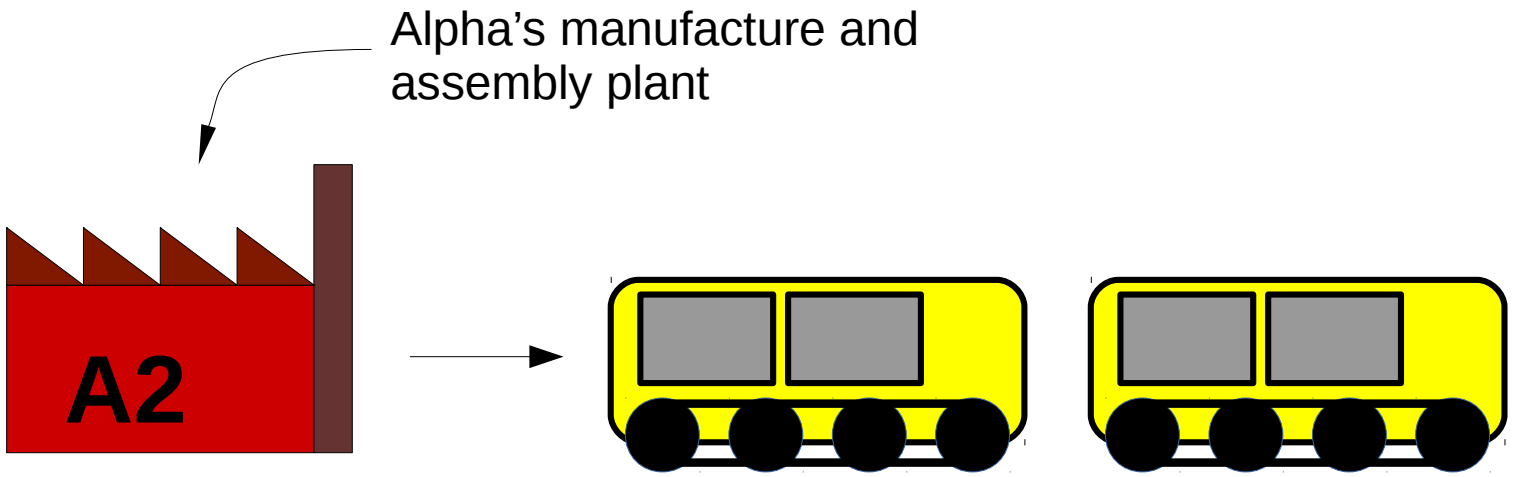
Component Servicing Use Case

A sample customer configuration

- Alpha corporation designs and oversees the manufacture of complex multi-part equipment, for example for heavy industry.



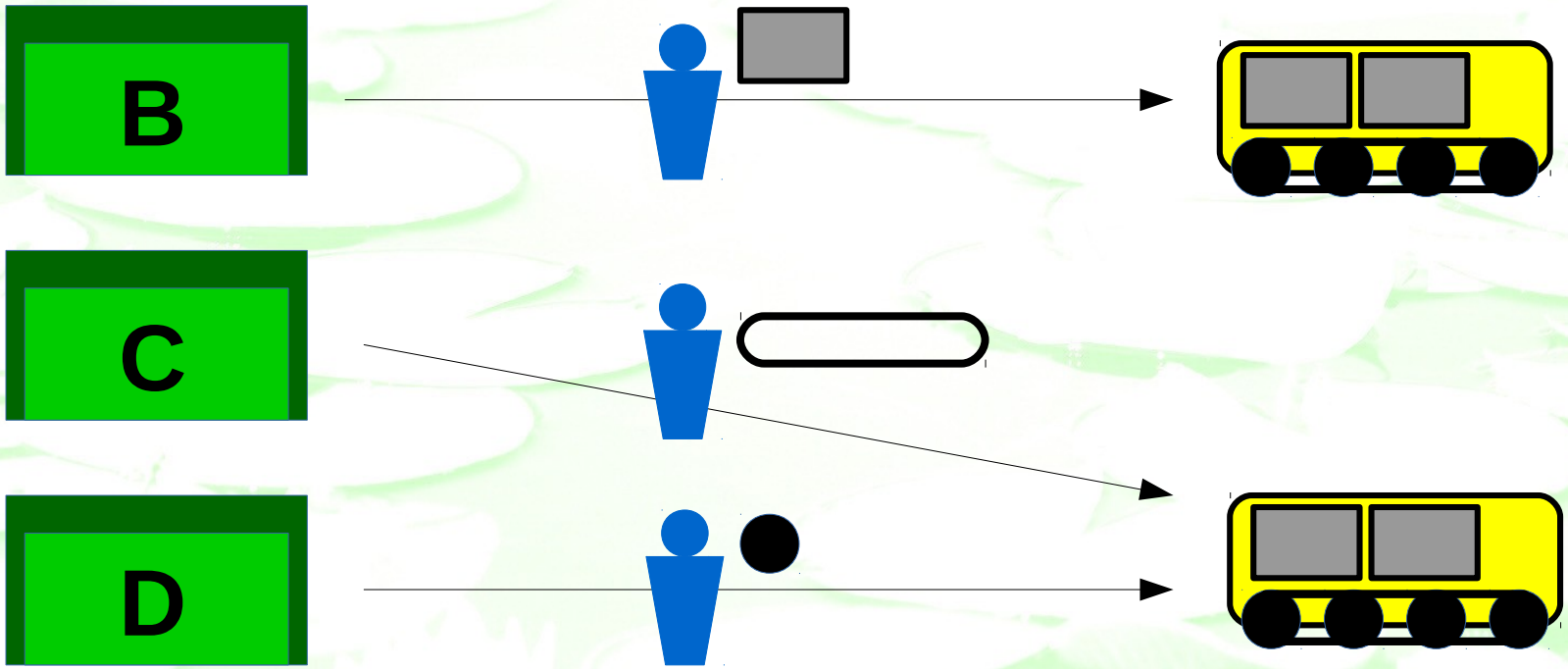
- The equipments components are produced and assembled in one of Alpha’s factories.



Component Servicing Use Case

A sample customer configuration

- The equipment is shipped to different remote locations and used heavily. Regular servicing and maintenance is required to comply with local safety rules and legislation.

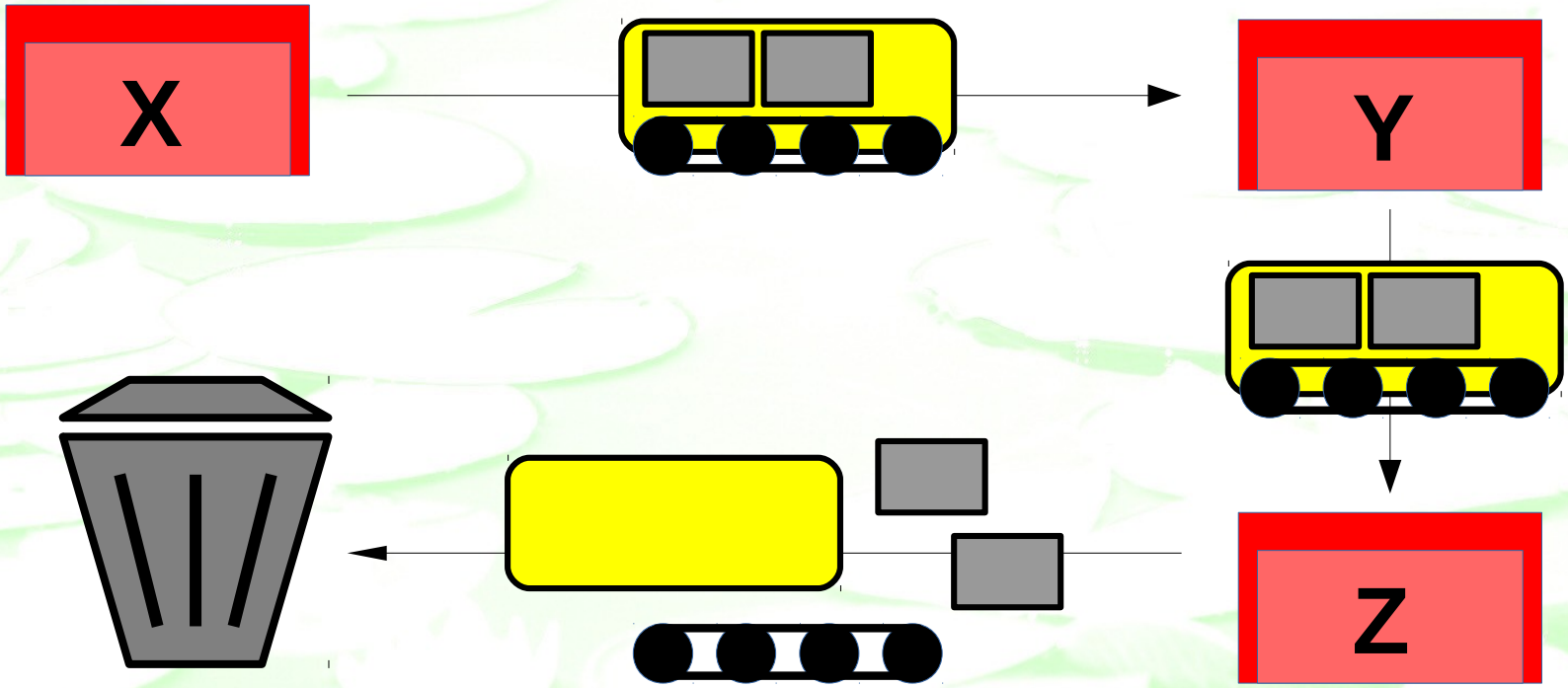


- Alpha contracts out equipment maintenance to authorized third parties, who use certified service engineers and approved replacement parts.

Component Servicing Use Case

A sample customer configuration

- Equipment may be sold from one corporation to another, and a record of the service history and provenance is vital.

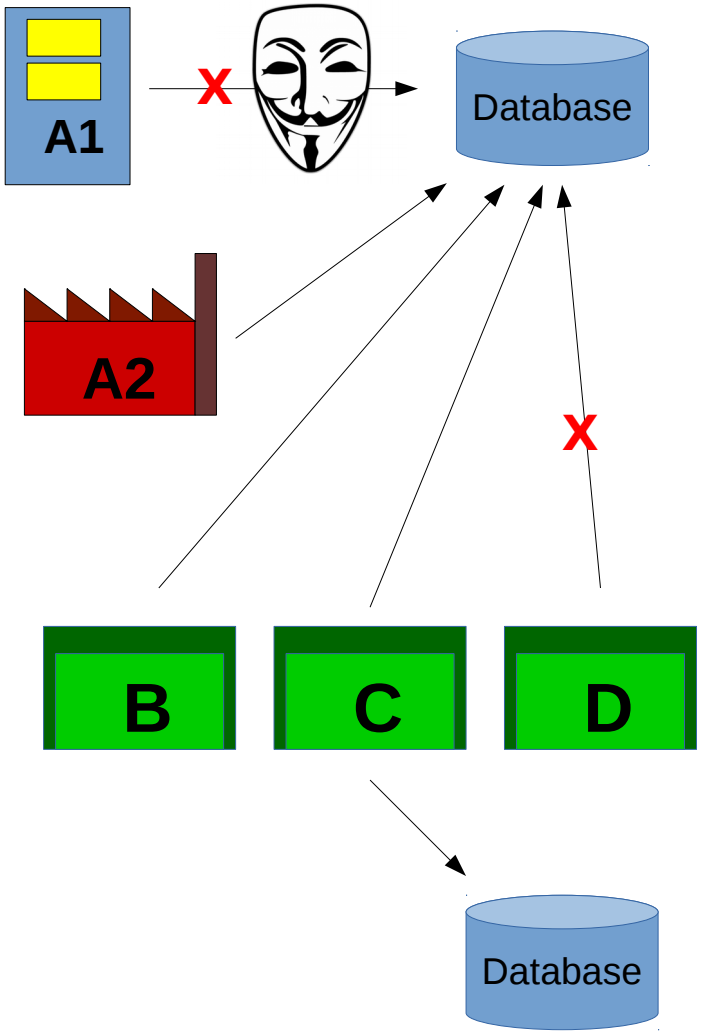


- At the end of the life-cycle the equipment is decommissioned

Component Servicing Use Case

The problem

- In a traditional setup, A will run a database tracking components, equipment, locations, service history and life-cycle.
- A denial of service attack on a database (or even database server farm) can put the operation serious risk.
- Records in any databases can be altered or deleted at a later date.
- Processes such as replacement part orders are manual, requiring human memory and intervention.
- Sharing access to the databases may be limited or need to be blocked by one or more of the participants at any time without notice.
- Compliance and third-party auditing, access permission granting and revocation, and interoperability between different systems all involve scheduling and IT infrastructure headaches.
- Disputes can be difficult to resolve at a later date, resulting in costly and protracted legal action.



Component Servicing Use Case

The solution

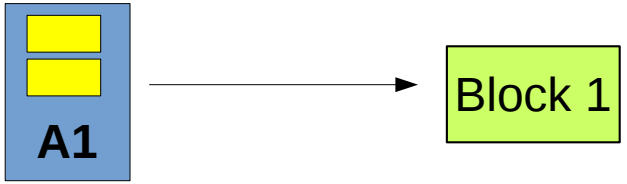
Component tracking and servicing on a blockchain is the solution:

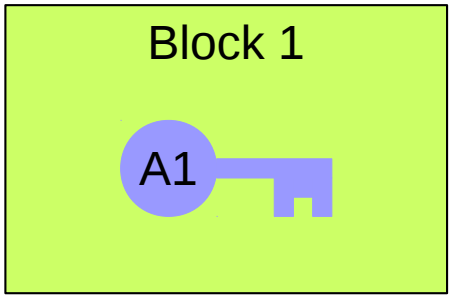
- Blockchains are distributed. Different nodes in different locations run the software powering the blockchain and keep a copy of the data. The more nodes there are:
  - the more backup copies of the data there are
  - the more backup “servers” there are
  - denial of service attacks become impossible
- Blockchains are tamper-proof. Once a component record has been on the blockchain for a short time it is locked down, and cannot be changed. Records cannot be disputed at a later date.
  - hacking the data records becomes impossible
  - “delete and deny” defenses become impossible
  - auditing is built in to the system by running a simple scan over the blockchain
- Blockchains can support “smart contracts”
  - components can have their required service and replacement history pre-loaded onto the blockchain using a “smart contract”
  - when the scheduled service date or wear-and-tear usage limit is reached, the system automatically triggers a service request
  - on completion of the service or replacement, a record is generated on the blockchain

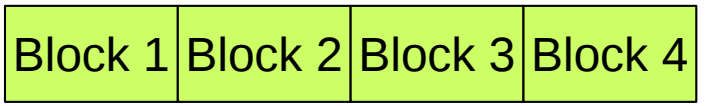
Component Servicing Use Case

Initializing the blockchain

- Alpha starts a “permissioned blockchain” with a “genesis block” - the first block on the chain
- Alpha runs the first blockchain node on a computer in its head office
- The genesis block contains the announcement message of Alpha’s first public key – this key will identify Alpha head office on the blockchain in future.
- The blockchain runs, and at regular intervals new blocks are added by Alpha’s blockchain node.



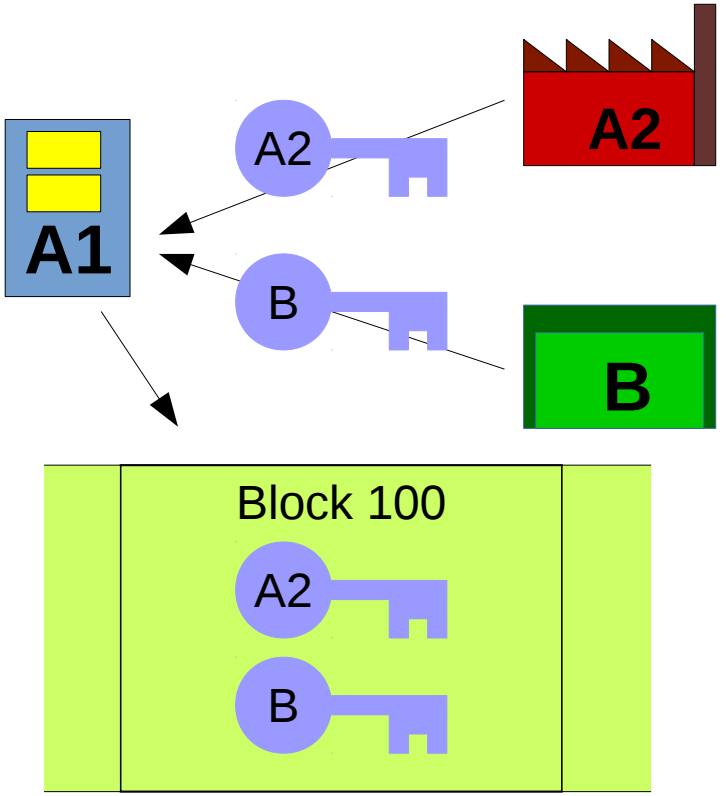




Component Servicing Use Case

Adding other participants

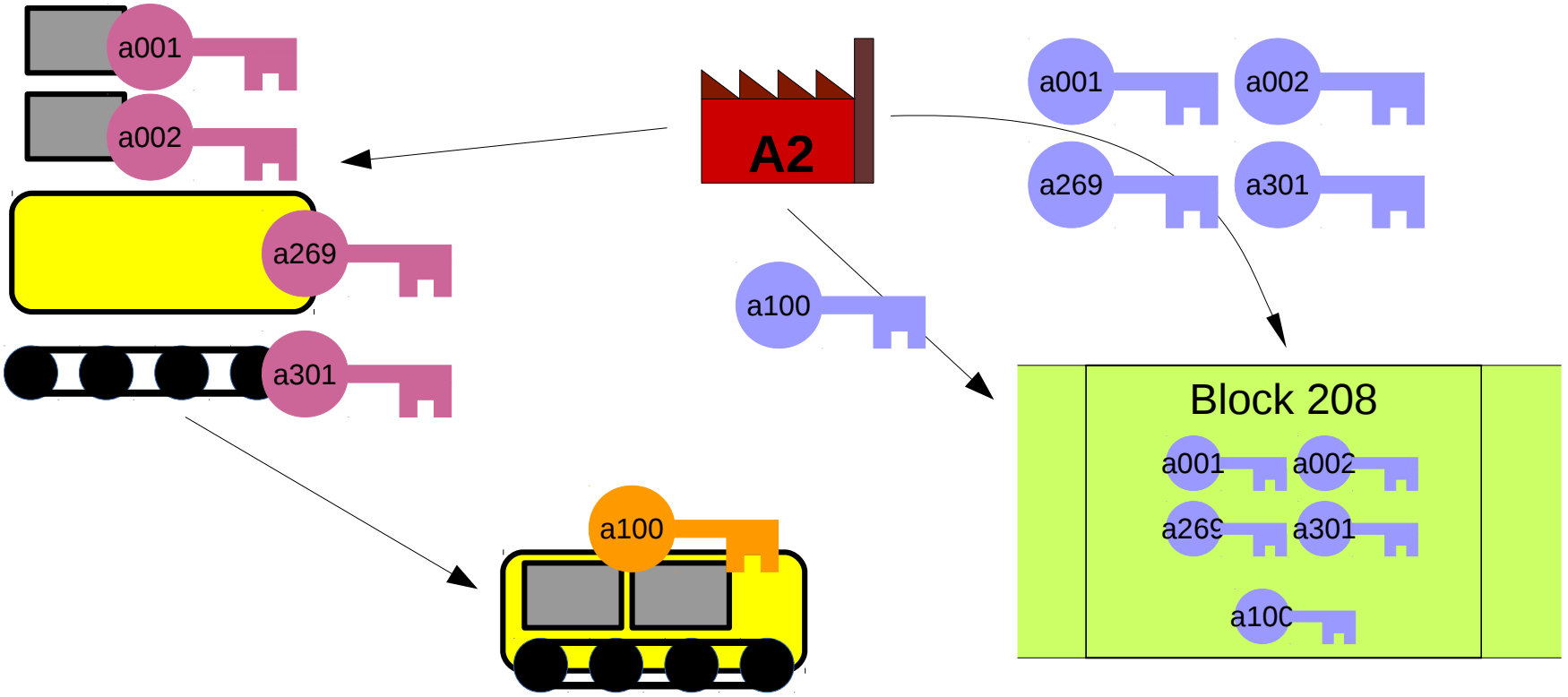
- Alpha's factory A2 and the contracted service company B generate public/private key pairs, and send the public key to A1, who announces them on the blockchain.
- Now A2 and B can run their own blockchain nodes to also add blocks onto the blockchain (they are in the blockchain peer-to-peer network)
- They can also announce new keys they create on the blockchain, for example when making parts or servicing equipment (more on this later...)
- Note that A1 does not know A2 and B's associated private keys, and they don't know A1's private key, so none of the participants can impersonate each other – a message on the blockchain signed with B's private key and verified with its public key has to have come from B.



Component Servicing Use Case

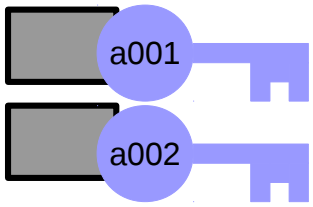
Keys for components

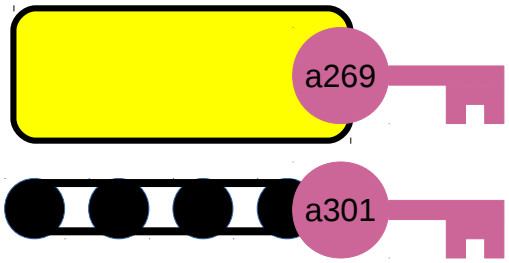
- A2 manufactures components. It creates a unique private/public key pair for each component, and announces the public key on the blockchain, along with the location of the part, which is its factory warehouse.
- Then the components are assembled to produce the final machine. Another key pair is generated for the finished product and is also reported.

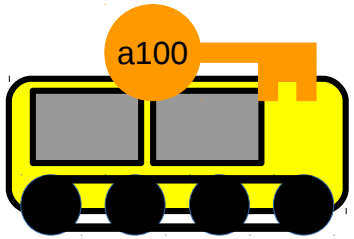


Component Servicing Use Case

Tagging methods





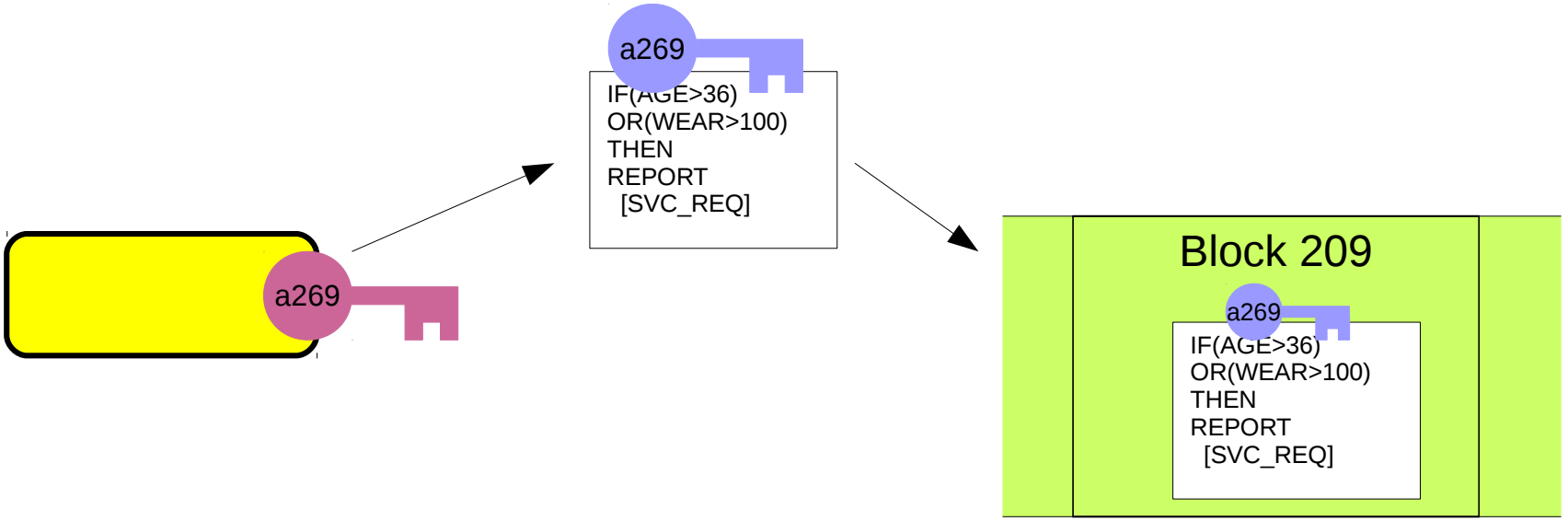


- Cheap components may be marked with a QR code of the public key for scanning, or a passive RFID tag. They cannot participate on the blockchain, and other RFID scanning devices read the key value and submit reports to the blockchain on their behalf.
- More expensive components can be marked with an active RFID tag, or a small IoT device with NFC/Bluetooth connectivity. They can “piggy-back” on nearby network connected devices to submit reports to the blockchain, and can sign challenge/response requests.
- Finished machinery may be fitted with a fully connected IoT device, with 3G connectivity to the network, onboard GNSS for positioning, and possibly even a lightweight blockchain node for participating in the generation of new blocks. An integrated RFID tag reader would allow this device to scan the complete machine for all its components, and would be able to detect changes made to those components.

Component Servicing Use Case

“Smart contracts” for components

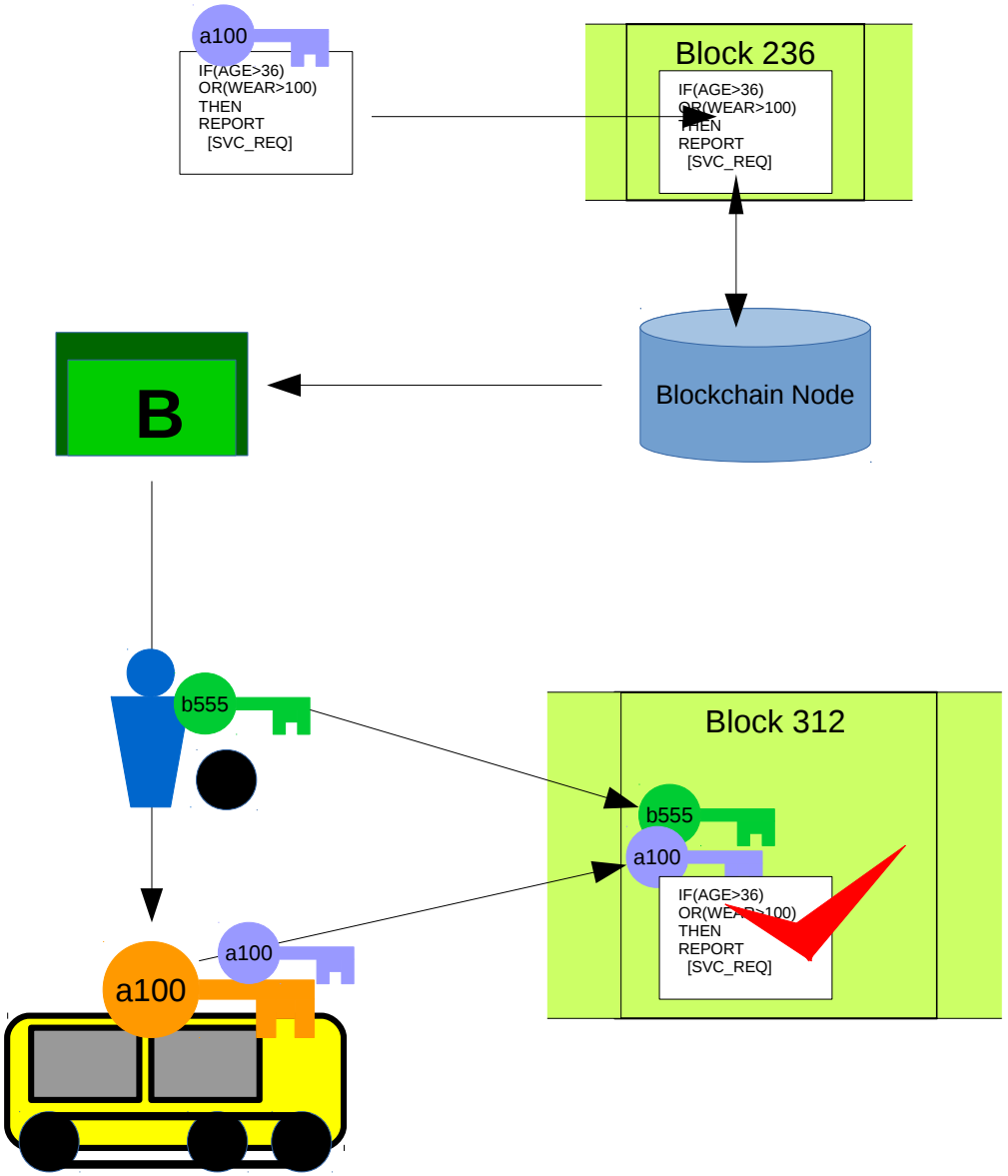
- When a component is logged on the blockchain by its public key announcement, a “smart contract” can be attached to the announcement.
- This is basically a simple program or script, which will be run by nodes maintaining the blockchain if certain conditions are met.
- Such contracts can trigger a service request, decommissioning, or part replacement order to automatically be placed with a service engineering company.
- The complexity of the automation provided depends on the design of the blockchain system.



Component Servicing Use Case

“Smart contracts” for components

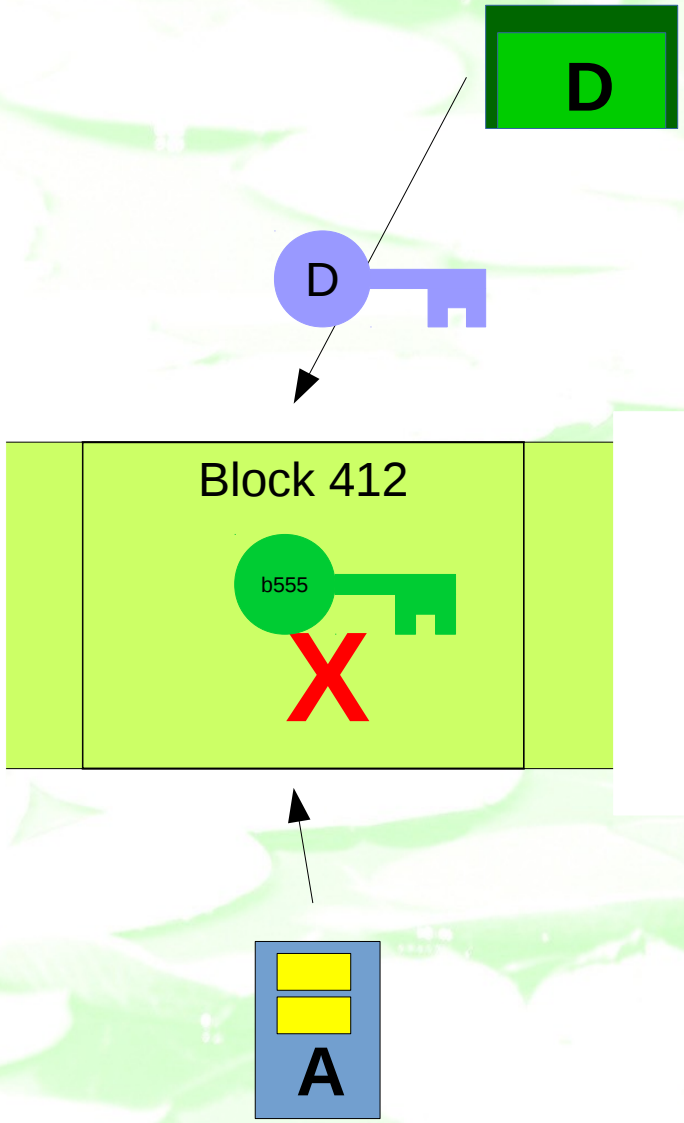
- A condition of the smart contract is met, and a node generating the current block sends out a service request to B.
- B sends a service engineer to the machinery in order to replace the worn-out part.
- The service engineer is certified on the blockchain by being issued with their own private/public key, which is stored in their smartphone or service tablet.
- When the engineer replaces the part, the service tablet and the machine’s central IoT device may send reports back to the blockchain to record the event. The contract is met.
- The service tablet may also be a node participating in the creation of new blocks...



Component Servicing Use Case

Adding and removing participants

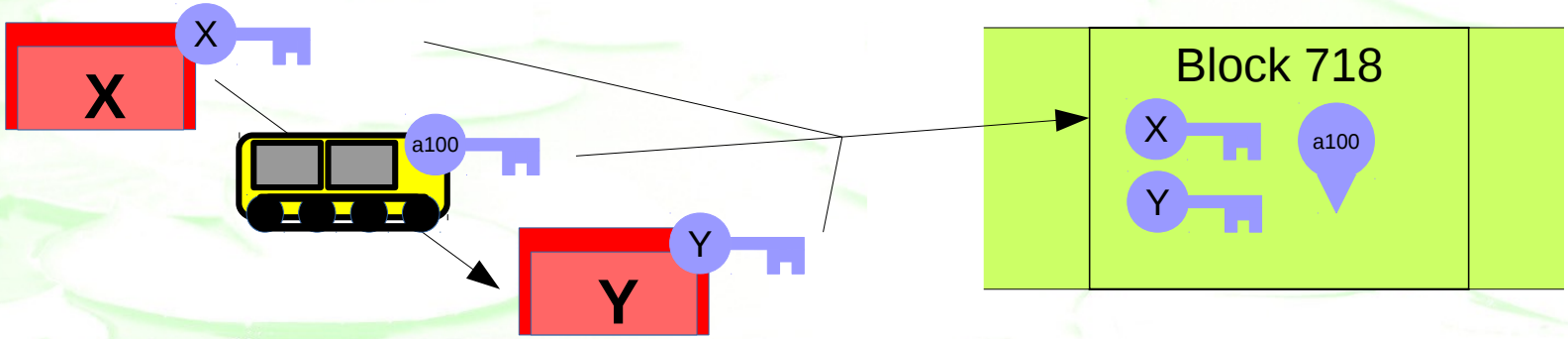
- B decides to revoke a service engineer's certification (e.g. because they left the company).
- They post a key revocation message to the block, signed with their private key, revoking key b555.
- They can do this because they issued and signed the key in the first place.
- All records made with key b555 prior to block 412 remain valid, however, and the records cannot be removed.



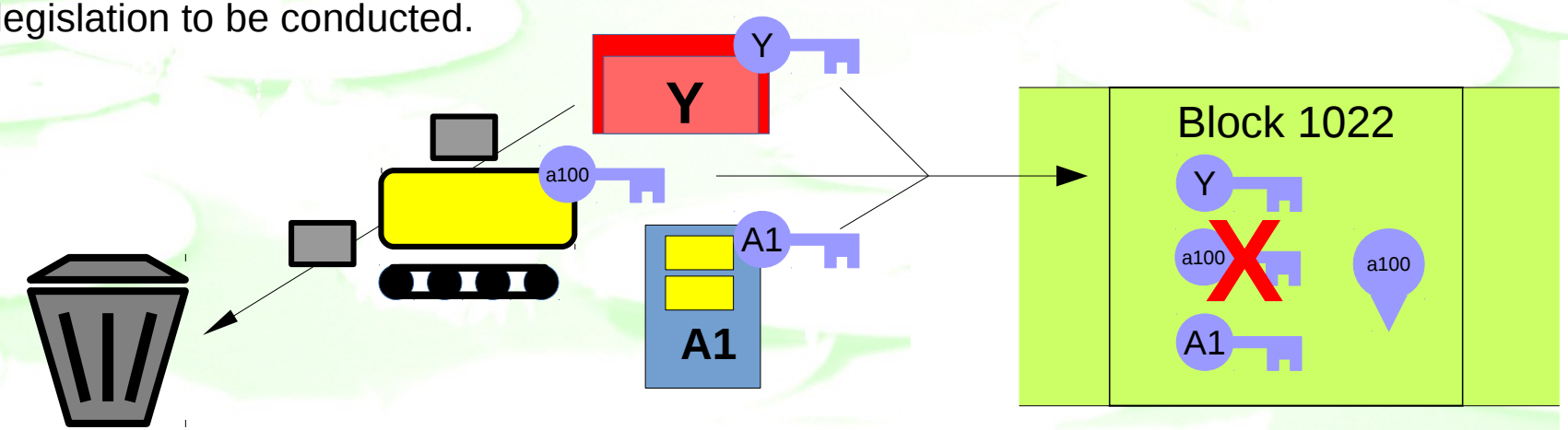
Component Servicing Use Case

Tracking a product life-cycle

- Company X is the owner of the machine, and sells it to company Y. The record of ownership and the location change of the machine is recorded to the blockchain, providing a future immutable record of its provenance.



- At the end of the life-cycle the equipment is decommissioned, and this event is also recorded on the blockchain, and possibly even witnessed and corroborated with a signature from the original manufacturer, A. This allows future trusted auditing of the application of environmental legislation to be conducted.



Component Servicing Use Case

Conclusion

- At the end of the process a complete record exists on the blockchain of all:
  - participants,
  - components,
  - their locations and journeys, who replaced what parts and where and when
  - transfer of ownership
  - final decommissioning
- The records cannot be altered or deleted afterwards
- Individual participants can track progress during transit, and review the data after delivery, alteration or transfer
- Only those directly concerned with or involved in the design, manufacture, repair or order fulfillment can create records relevant to that stage of the process
- There is no central server or central point of failure