Chapter 4
Example Use Cases

To convey a more concrete picture of applications of blockchain, this chapter presents four exemplar use cases which illustrate some of the techniques and considerations discussed in the previous chapters. These use cases are also used as running examples throughout the book, but their details are not strictly necessary for understanding later parts of the book. For every use case, we give a brief background and describe their key non-functional requirements.

4.1 Agricultural Supply Chains

In manufacturing, retail, and agricultural industries, supply chains are critical in the movement of goods and services across organizational boundaries. Supply chain contracts are complex, dynamic, multiparty arrangements, with regulatory and logistical constraints. They often cross jurisdictional boundaries. The information exchange in a supply chain can be as important as the physical exchange of goods. For example, customs inspections would not start until both the physical goods and the information about those goods are present. Confidence in supply chain documentation can expedite customs and biosecurity processes, reduce risk and insurance costs, and be used as leverage in trade finance. Payments are made between parties at many points in the supply chain.

For agricultural food products, being able to tell where ingredients were grown and how products were processed and distributed can be important in establishing confidence in food safety, creating and building high-quality brands, reducing fraud, and improving supply chain efficiency. There are many stakeholders in an agricultural supply chain, ranging from producers to transport providers, sorting/processing facilities, wholesalers, distributors, retailers, and consumers. In international supply chains, there are also stakeholders related to customs and biosecurity. A simplified configuration of some stakeholders and functions is shown in Fig. 4.1.

X. Xu et al., *Architecture for Blockchain Applications*,
https://doi.org/10.1007/978-3-030-03035-3_4

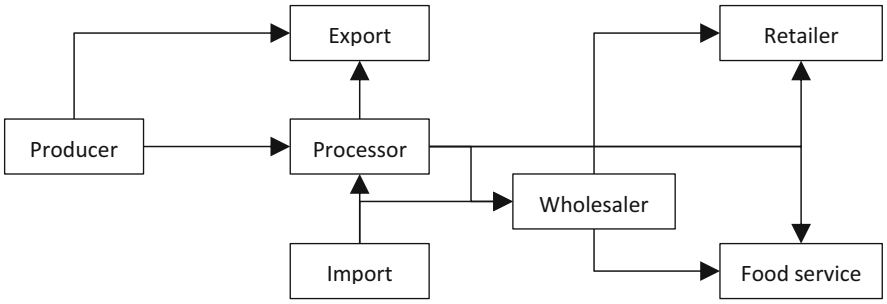

Fig. 4.1 Stakeholders in a simple agricultural supply chain. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

The information systems supporting supply chains normally reside at the individual supply chain participants and are integrated to varying degrees, i.e. from no digital integration to machine-readable barcodes that can be understood by a number of participants, through to full system integration with digital message exchanges.

4.1.1 Key Non-functional Requirements

- Interoperability: A huge challenge in logistics is to coordinate information exchange across the many different kinds of goods, modes of transport, and information systems. Individual shipments can be aggregated into larger consignments, which means tracing information about the status of goods can require integration of different interlinked information sources.
- Latency: The exchange of physical goods must sometimes wait upon exchange of documentation associated with the delivery. Information exchange should not introduce significant additional delays at these points.
- Integrity: Supply chain quality and provenance require that information about goods and supply chain events cannot be falsified or created without proper authority.
- Confidentiality: Some information in supply chain documentation should be held commercial-in-confidence. Even metadata can expose aggregate trade flows which can be commercially sensitive. However, because of long supply chains and the use of subcontractors, parties' interests in information about supply chain events may extend beyond the parties directly involved in that event. Balancing transparency and commercial confidentiality is a complex business model problem.
- Scalability: There are many supply chain processes in progress at any time across a large number of different parties. Each process instance creates a large number of events, although not all events are relevant to all participants. A system must scale to handle the total throughput of transactions, with parties using resources in proportion to their level of involvement in the process.

4.1.2 Conventional Technology

Traditionally, supply chain information is recorded separately by each entity in the chain. Each participant only sees the information they are a direct party to. As supply chain systems have become more digitized, information sharing has become more common. Standards such as GS1's EPCIS (Electronic Product Code Information Services) define uniform schemes for representing supply chain events. This can help parties in a supply chain to record and exchange information.

Figure 4.2 depicts a design for a supply chain system using EPCIS and other data with conventional technologies. All EPCIS data is sent to a central event aggregation server for an agreed portion of the supply chain. A group of supply chain participants agree on a trusted party to operate and control access to the aggregation server. Note that this design would be an advance over many current supply chain systems, but it has been implemented in some industrial settings. Note also that the centralized server creates a risk as a single point of failure, either for operational reasons or for business reasons. (Business reasons may include complete business failures or perhaps merely unfavourable changes in pricing or terms of use.)

Supply chain events are not the only type of information that needs to be exchanged. Other documents may include letters of credit, bills of lading, booking confirmations, arrival notices, container releases, terminal load lists, delivery orders, tax invoices, and so on. These other types of documents are normally kept locally to the systems of the different supply chain participants and exchanged directly using

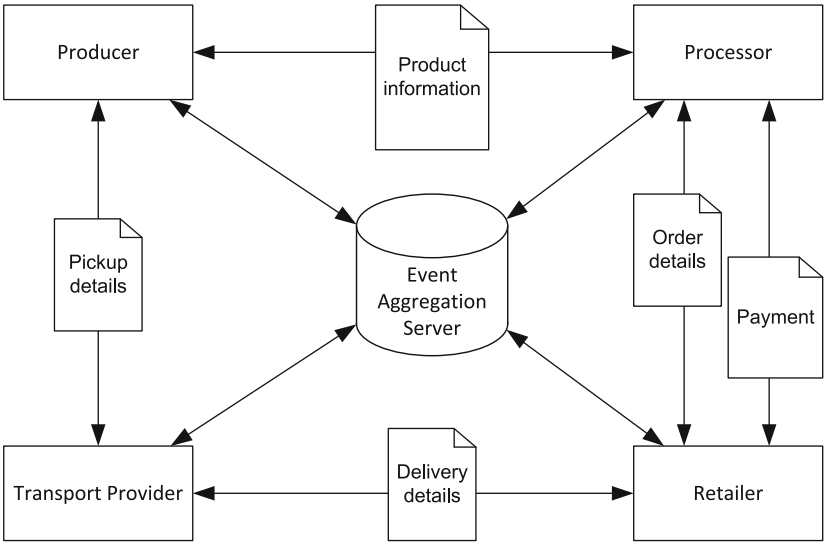

Fig. 4.2 Model of supply chains using conventional event aggregation server and point-to-point integration. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

point-to-point integration between parties. Currently, it can be hard to guard against fraud that uses forged or tampered documents.

4.1.3 A Blockchain Solution

One possible alternative solution using blockchain is to control the execution of the process of a supply chain using smart contracts. A group of participants that want to implement a shared supply chain process first agree on a design for the collaborative process that regulates how their interactions should take place. The controls for this process are implemented using smart contracts, and the participants coordinate the progress of that process by calling those smart contracts in turn. The smart contract can enforce the process as follows. First, it can reject messages if they arrive at the wrong point in the process. Second, messages are only accepted from the participant who is authorized to send them. Third, conditions can be specified within the process model and can be executed in smart contract code directly. So particular process branches will be automatically activated when their conditions are met.

Consider an example: containerized export of wine from a rural Australian producer. This starts when the producer initiates a shipment and ends when the container is on a ship. One process instance deals with exactly one container, and once the container number is assigned, it can be used as an identifier of the process instance. Figure 4.3 shows the process model.

These smart contracts can be generated automatically from process models, as we discuss in Chapter 8. In the resulting system, the supply chain participants interact with each other by sending messages through the blockchain. To facilitate interaction through blockchain, so-called trigger components act as bridges between the blockchain and enterprise applications. The trigger can translate conventional service calls to blockchain transactions and vice versa. This can keep the implementation cost relatively low. For message formats, we can use the same standards as in the conventional design, i.e. GS1 EPCIS.

4.1.4 Non-functional Property Discussion

Interoperability Both designs use the GS1 EPCIS standard for events. The first design requires point-to-point integration between any two participants for other documents. Extending the supply chain to a new participant requires integration of that participant's system with all participants that need to exchange documents directly with the new participant. The second design requires the same amount of integration initially: the data formats also need to be agreed upfront. However, each new participant only needs to integrate their systems once, with the blockchain process, and thus the overall integration burden is reduced.

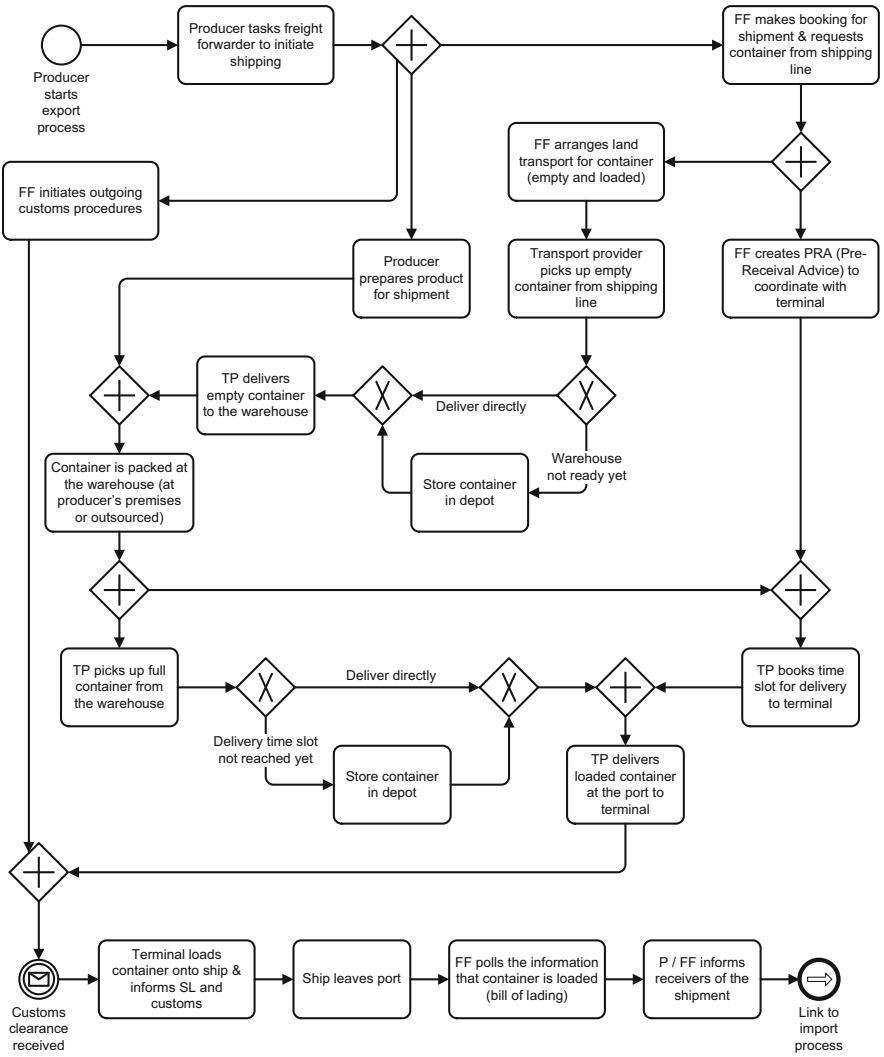

Fig. 4.3 Process model of an agricultural export supply chain process. *FF* freight forwarder, *TP* transport provider, *P* producer. Notation: BPMN. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

Latency Supply chains typically involve the physical movements of goods, so many latency requirements on information transfer are usually on the order of minutes to hours. Neither of the designs should suffer from latency exceeding these time frames. However, at points of handover of goods, there may be low latency requirement for confirmation of receipt of goods. Commit times on public blockchains are likely to be too long for this, but it may be possible to instead

provide cryptographically signed receipts off-chain, with the delivery agent able to lodge those to the blockchain at a later time.

Integrity The first design relies on a trusted party to operate the aggregation server and is subject to the possibility of manipulation with a low chance of detection. Integrity is a strong inherent feature of blockchains: information captured as part of committed transactions would be exceedingly hard to change. If large blocks of data (such as photos or video) need to be stored, this could be done off-chain, with integrity preserved by storing a cryptographic hash of this data on-chain. This allows detection of alterations or corruption of the off-chain data, but increases design complexity.

Scalability In both designs, each party has to deal with the scalability of their own enterprise applications, which we do not discuss here. Instead, we focus on scalability of the components shared by all parties. In the first design, this is the central aggregation server. If all participants publish all event data for item movements, this might become a bottleneck. There are many design options available to address scaling of web-based centralized information systems, including filtering to only publish events that are relevant for other parties and using load balancing services to federate data access across multiple aggregation servers.

In the second design, the component shared by all participants is the blockchain. Scalability of reading from the blockchain can be good, since each participant can hold their own full copy of the blockchain. For writing new transactions and smart contract method calls, scalability is currently limited on public blockchains. For this design, we propose using a *consortium blockchain*, where transaction volumes can be controlled and where other technical options for block formation and consensus are available to improve performance. As with the first design, only relevant events should be stored on-chain. In the second design, communication is also limited to the messages exchanged as part of the collaborative process execution. Throughput scalability can be achieved by careful design and performance tuning. As discussed in the previous chapter, specific types of blockchains that do not use Nakamoto consensus have been designed for private or consortium blockchains with high scalability requirements.

Confidentiality Confidentiality requirements for supply chain data are not the same across industries or participants. This affects both designs: for a specific supply chain and a specific set of participants, the confidentiality requirements need to be formulated and analysed, and potentially the design needs to be adapted accordingly. The main trade-off is between the benefits of sharing data within the group of collaborators—visibility and cross-party optimizations are impossible without that—retaining confidentiality between competitors where needed. Supply chain information can be commercial-in-confidence. This may include the identities of participants, trade volume, prices, and delivery times.

While it is possible to restrict access to the aggregation server in the first design and the consortium blockchains in the second design, it should be expected that

for some supply chain roles multiple competing participants have access to the same system. Even a private blockchain does not protect commercial-in-confidence information. Unless the supply chain is entirely vertically integrated within one organization, competitors will be sharing access to information on the blockchain. The only way to prevent that is by setting up a separate aggregation server or blockchain for each group of parties. That is, switching transport providers would require setting up a separate system, which would not only be tedious and resource-intensive, but would also severely hamper the analysis of supply chain data across specific instances. Alternative distributed ledger technologies, such as R3's Corda or Hyperledger Fabric, natively support the creation of separate ledgers for related parties, e.g. through Fabric's channels. However, they still suffer from the second issue: the lack of visibility hampers global analysis and optimization.

Data stored on a blockchain is readable to all participants of that blockchain. Confidential data can be encrypted, and keys can be exchanged between supply chain participants so that only the 'right' group of participants can decrypt that data. However, this requires off-chain key exchanges and diligent handling of keys. Moreover, normally encrypted data can itself not be processed by the blockchain or its smart contracts. Thus, transfers of assets that are managed by the blockchain cannot be encrypted; and encrypted data cannot be transformed or actioned by smart contracts. New sophisticated cryptographic techniques such as homomorphic encryption and zero-knowledge proofs allow various kinds of computation or transaction validation to be performed on encrypted data, without decrypting it. These techniques are being explored for use in blockchain platforms and may provide an alternative treatment for this issue.

Finally, a confidentiality concern can arise from metadata, not just data inside the transactions. For example, the volume of interactions between parties may reveal trade volumes. It would be possible to create new account addresses for each participant and each new process instance, but the flow of assets may still be used to infer relationships between addresses, revealing aggregate trade volumes. Dummy transactions might be used to attempt to hide this. Such protection mechanisms can help, but may erode the benefit of using a blockchain. These trade-offs require careful consideration.

4.2 Open Data Registry

Registries are authoritative collections of information, usually managed centrally, often by government agencies. A registry holds information about a class of entities. Examples of such entities include individuals, businesses, species, and organizations. In Australia, familiar registries include the immunization registry, the business name registry, and land title registries. There are also well-known international registries such as the Domain Name System (DNS). Some government registries are described as 'public' and can be queried by individuals. However, query access to these registries may be limited to prevent attempts at republishing or

data mining. Unfettered data mining could threaten commercial or personal privacy and is often restricted using regulatory policies, query rate limits, and user access controls.

Some government registries contain periodically published open data. In Australia, these are published through `data.gov.au`.[1] In this use case, we specifically consider the use of blockchains for managing an open data registry of datasets, data sources, and data analytics services. This means we do not consider confidentiality or privacy issues for this use case. Blockchains provide transparency about their entire transaction history to all processing nodes. In a public blockchain, this means that the information is openly published. It is possible to run a private blockchain hidden behind a web service or other interfaces. This could limit access to the registry in a way that satisfies an appropriate access policy. However, many of the benefits of using a blockchain would be foregone in such an architecture.

For open data, the major stakeholders are data providers, data consumers, and the data registry. Data providers may include government agencies, research institutes, universities, and companies. Data providers record metadata about their datasets on the data registry and make their data available on their websites. Data consumers query to discover datasets in the data registry based on the metadata. They can then download the datasets from the data providers for analysis.

4.2.1 Key Non-functional Requirements

- Integrity: each data provider should only be able to create and change registry entries for their own datasets.
- Availability: there should be high likelihood of being able to access the registry when desired, for both data providers and data consumers. This particularly applies to national public registries, which form the basis for many other services that utilize the data from the registries.
- Read latency: data consumers may need to repeatedly query the registry while browsing and searching for relevant datasets. This may be done programmatically from a graphical user interface and so should have low latency.
- Interoperability: a registry may reference other registries to reduce duplication and errors.
- Ease of integrating new data providers: to grow the network effects of the registry as a data portal, it is important to have low barriers (time, cost, and administrative burden) to add new data providers to the registry.

[1] https://data.gov.au.

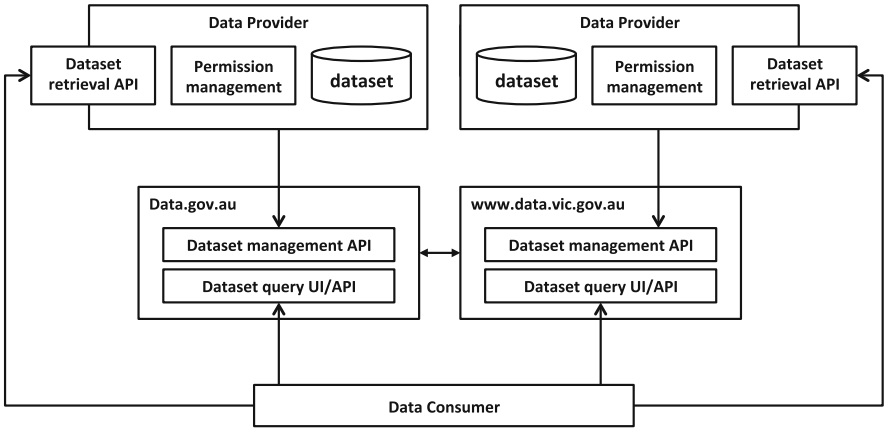

Fig. 4.4 Design for a registry using conventional technologies, operated by a single agency. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

4.2.2 Conventional Technology

Data portals such as `data.gov.au` implement a dataset registry using conventional technologies such as CKAN.[2] For each portal, the CKAN software is run and managed by a single government agency. Data consumers interact with a registry to discover datasets but retrieve datasets directly from data providers. The data providers may perform some permission management for data access independently. An illustrative high-level design is shown in Fig. 4.4.

In the CKAN ecosystem, datasets in different CKAN repositories refer to each other by importing metadata from each other.

4.2.3 A Blockchain Solution

We consider a design which replaces the registry with a public blockchain. In this design there is no single agency that operates the registry. Instead the data providers independently record metadata on the public blockchain and perform their own permission management and access control for their datasets independently. Note that there may still be an agency leading governance for the registry. In this design, data consumers are required to interact directly with the blockchain, rather than with a consumer-facing user interface or API. Those consumer interfaces may be provided by commercial or personal systems. An illustrative high-level design is shown in Fig. 4.5.

[2] https://ckan.org/.

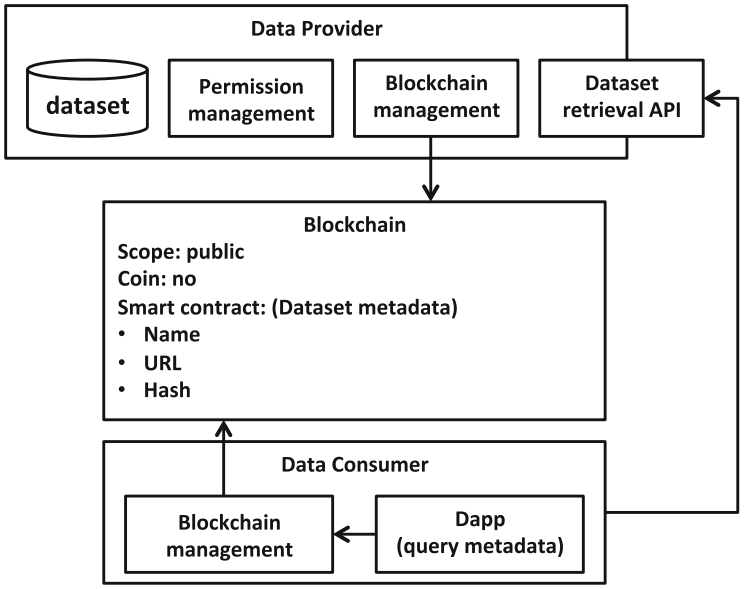

Fig. 4.5 Design for a registry using a public blockchain. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

4.2.4 Non-functional Property Discussion

Integrity The conventional design relies on a registrar to create registry entries on behalf of data providers. New registry entries are validated by the registrar. In the blockchain-based design, registry entries can be created directly by the data providers, using their private key. Registry entries are validated by smart contracts checking data integrity conditions, and all transactions are validated by all processing nodes in the blockchain network. Data consumers hold a local copy of the blockchain, through which they access the registry.

Availability In the conventional design, the data registry system is a single point of failure for availability for all stakeholders. In the blockchain-based design, there is increased data redundancy which can improve read availability for data consumers. For the open data use case, write latency is not critical, which allows satisfactory service availability despite possibly lower write availability than the conventional design.

Interoperability In the conventional design, the datasets in different CKAN repositories refer to each other by importing the metadata from each other using standard formats but optionally with customer-defined fields. The blockchain-based design has a uniform technical infrastructure. The shared smart contract validation

rules will reduce the likelihood of incompatible data formats, which means different registries will be more consistent with each other.

Read Latency Reading in the conventional design is performed through a remote API over the Internet. Compared with the blockchain-based design, this is slower: a local blockchain node is collocated with the consumer's query interface, and reading is done locally at high speed.

Ease of Adding Providers In the conventional design, new data providers are added by the central registrar using registry backend services. In the blockchain-based design, new providers can join by independently creating a new public/private key pair. Authentication of their public key could be certified by a registrar on the blockchain or separately off-chain. Data providers must integrate with the blockchain, and should ideally run a blockchain node.

4.3 International Money Transfers

Many workers in Australia regularly send money back to their families overseas. These flows of cash constitute up to about 10% of GDP in some developing countries (and even 27% in Tonga and 20% in Samoa). Thus, high remittance costs have important implications on socio-economic development of these countries. Remittances are low-value, high-volume payments. However, remittance costs in Pacific Island countries are among the highest in the world. For example, to send $200 from Australia to Vanuatu costs $33.20 and $28.60 to Samoa.

There can be many parties involved in the chain of transactions made for these payments, and there is sometimes little transparency on the total cost of exchange rates and fees. Remittance payments can also be complicated by the difficulties of satisfying AML/CTF (Anti-Money Laundering/Counter-Terrorism Financing) regulation, especially where the receiving party may not have a bank account. These transactions can have high latency, with transaction times ranging from less than 1 h to 5 days.

In this use case, stakeholders include remitters, beneficiaries, and different types of financial institutions, including banks and Money Transfer Operators (MTOs). We consider the stakeholders and functions depicted in Fig. 4.6.

To be able to complete a remittance payment, both the remitter and beneficiary initiate a relationship with the financial institution. A Know Your Customer (KYC) process is conducted by the financial institution. The remitter pays a financial institution from the remitting territory who transfers the money across the border. Another financial institution from the beneficiary territory receives the money, exchanges it to local currency, and disburses it to the beneficiary. Prior to the completion of the exchange, and depending on the amount of money transferred, transactional level Anti-Money Laundering (AML) and Counter-Terrorism Financing (CTF) checks required by regulators in either territory (and in any financial

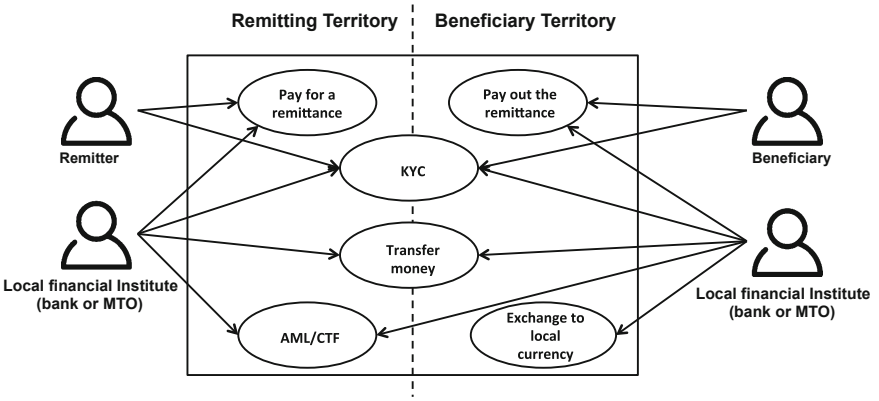

Fig. 4.6 Stakeholders and functions for remittance payments. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

institutions in intermediate territories) may be performed on the identity of the remitter and beneficiary, perhaps including assessment of the purpose of the transfer.

4.3.1 Key Non-functional Requirements

- Transaction latency: completing a remittance payment should ideally be instantaneous, or at least take place comfortably within the context of human interaction with a physical kiosk or web form.
- Cost: the total cost of remittance should be a low percentage of the transaction value.
- Cost transparency: the total expected cost including fees and exchange rate should be visible to participants.
- Controlled confidentiality: for regulatory compliance, all required AML/CTF checks must be performed, but appropriate levels of commercial confidentiality must also be maintained.
- Barriers to entry: increased competition can drive lower costs and greater service innovation, but this requires low barriers to entry (cost, time, and regulatory burden) for new remittance service providers.

4.3.2 Conventional Technologies

The process for banks depicted in Fig. 4.7 starts when the remitter deposits money into their bank. The remitter's bank then initiates a SWIFT wire transfer to send the money across to the beneficiary bank, possibly through several intermediary

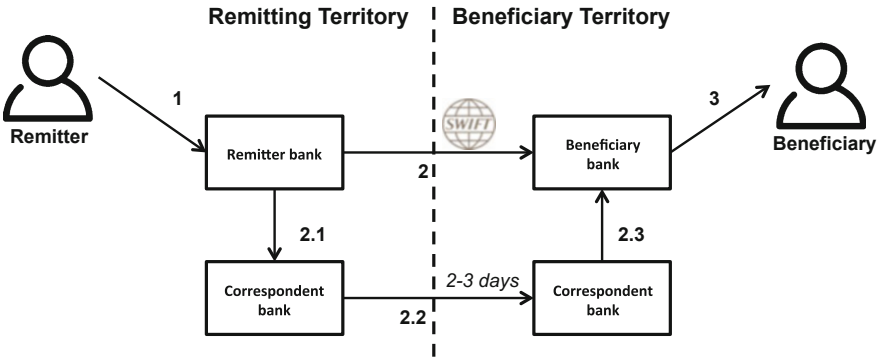

Fig. 4.7 Remittance through banks. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

correspondent banks. It can take 2–3 days for the money to be sent. The receiving bank then informs the beneficiary's bank that the money in the foreign currency has arrived and transfers the local currency equivalent to the beneficiary's bank. Finally, the beneficiary's bank disburses the local currency to the beneficiary.

Another widely used way to do remittance is through a Money Transfer Operator (MTO), as depicted in Fig. 4.8. In this case, a remitter uses either cash or other payment instruments to pay the MTO. Once a group of payments is received, the remitting MTO pools all money into a single transaction. The MTO also prepares a file with instructions on breaking down the remittance to individual orders and sends the file to the beneficiary MTO. Then, the money is transferred by the MTO to its foreign bank as a normal international transfer, as per the above process. The bank

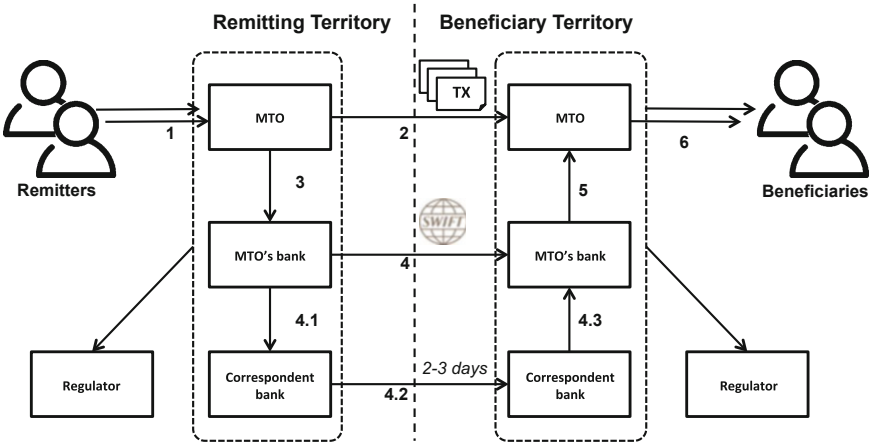

Fig. 4.8 Remittance through MTOs. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

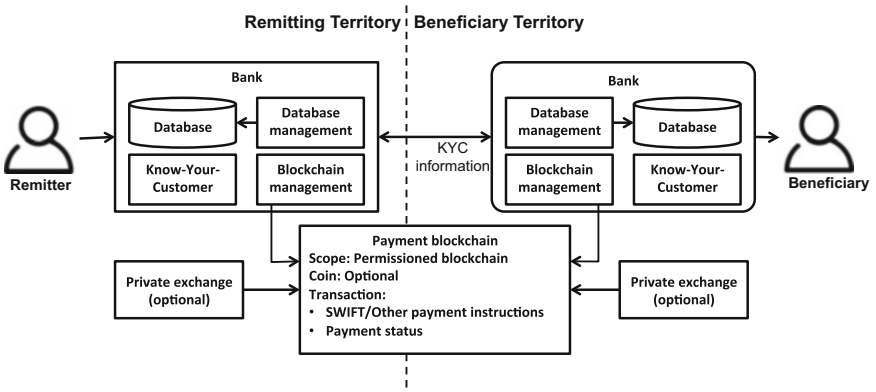

Fig. 4.9 Payment through blockchain. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

charges the MTO once for all the remittances. When the beneficiary MTO receives the money, it distributes it according to the instructions received earlier.

4.3.3 A Blockchain Solution

Banks, financial institutions, and MTOs could join a private blockchain to enable real-time settlement, as depicted in Fig. 4.9. Apart from speeding up money transfers, blockchain could also help banks to operate continuously, 24 h a day. The on-chain portion of the design can include SWIFT instructions or other payment instructions and the payment status. The native currency of the blockchain can be used as an intermediary currency by banks to facilitate foreign exchange. KYC and risk information, fees, and foreign exchange rates are exchanged through conventional means, off-chain.

When Bitcoin is used, this is sometimes called 'rebittance'. Some companies use Bitcoin directly as an intermediary currency for foreign exchange. The underlying Bitcoin layer is invisible to end users. In this case, every remittance has a corresponding transaction recorded on the Bitcoin blockchain. Other companies maintain a separate blockchain to facilitate settlement among branches and anchor their blockchain with the Bitcoin blockchain as a way to leverage Bitcoin's immutable, independently auditable ledger.

4.3.4 Non-functional Property Discussion

Transaction Latency The systems following conventional designs can result in time-consuming transactions, e.g. depending on the route, specifically the number of

correspondent banks involved. End-of-day batch processing causes delays of up to 24 h, and time zone differences can cause delays of up to 24 h. The blockchain-based design enables real-time processing with latencies that vary from seconds to hours, depending on the blockchain. For example, on Bitcoin the latency averages around 1 h if 6 confirmation blocks are used; using public Ethereum with 12 confirmation blocks would on average take around 3 min.

Cost In both designs, remitting banks charge transaction fees, and liquidity providers charge via the spread on foreign exchange (FX) rates. There are also correspondent bank fees in the conventional design.

Transparency In the conventional design, each bank in the payment chain is aware of its own actions, but some KYC information is transmitted through the chain of correspondent banks. How FX spread is calculated and what will be charged in fees is not always predictable. In the blockchain-based design, a common shared view of the payment status enables real-time fraud analysis and prevention. On Bitcoin, regulators and others can access historical data in the blockchain but would need additional information to know how to interpret the pseudonymous addresses and the identities of senders and recipients.

Controlled Confidentiality In the conventional design, KYC regulatory compliance requires costly technology capabilities and complex business processes. There is substantial duplicated effort between banks and other financial institutions. The blockchain-based design replaces intermediary banks with a blockchain to provide a shared record of payments and KYC checks and thus may simplify regulatory compliance along the payment chain. Some automated and real-time compliance checks may be available on-chain using smart contracts, depending on the blockchains used.

Barriers to Entry The conventional design requires participants to have banking or financial services licenses, and business relationships with correspondent banks. The second design requires new technology development and integrations, but some existing transaction standards can be reused. Interaction between separate proprietary blockchains would require inter-ledger protocols. Public blockchains have low barriers to entry for new participants, but regulatory or banking constraints for digital currency exchanges apply to end-points within countries.

4.4 Electricity Contract Selection and Continuous Reporting

Electricity consumers may change their electricity retailers based on their usage and current offers from electricity retailers. Typically, there are conditions associated with the contract between the electricity consumer and the retailer. For example, retailers may offer discounts if bills are paid on time or may require exit fees if the contract is terminated ahead of time. Some retailers also allow flexible payments, such as weekly, fortnightly, or monthly payments. There are two participants in this

scenario: the end user and the electricity retailer. We assume that a smart meter is attached to the end user's place of supply and that this smart meter is connected to the network and can digitally sign messages using a private key.

4.4.1 Key Non-functional Requirements

- Integrity: The monthly usage of an electricity consumer is an important criterion for consumers to select an electricity retailer and for electricity retailers to make special offers. Therefore, accurate records of usage are important to prevent deception between the parties.
- Privacy: Current and historic electricity usage data can be used to infer private information—researchers have even shown that accurate high-frequency smart meter readings allow identifying which movie an end user is currently watching. More coarse-grained data could also be used by burglars to find out when someone is on vacation. Therefore, usage data should only be shared at the discretion of the end user.
- Transparency: Historical electricity usage, perhaps associated with previous electricity retailers, could be used by a prospective retailer to customize new special offers. As discussed above, we assume that consumers are able to authorize the sharing of their usage information with other parties.

4.4.2 Conventional Technologies

In conventional environment, every electricity retailer uses its own bespoke system to maintain customer data and smart meter information. Historical electricity usage is not normally shared among electricity retailers. Payments are made through traditional banking systems.

4.4.3 A Blockchain Solution

In this blockchain-based design, we propose using a consortium blockchain as a platform to track historical electricity usage of every smart meter and to provide payment services. The architecture of the solution is shown in Fig. 4.10.

When a user wants to find a new retail supplier, they create a retailer selection smart contract on the blockchain, against which retailers can bid. To bid, retailers create a smart contract offer from an offer template. The offer contract is defined using variables such as start time, end time, energy level, level price, service fee, and charge date. Transactions listed on the blockchain provide a history of usage associated with smart meters and users. This information can be accessed by retailers

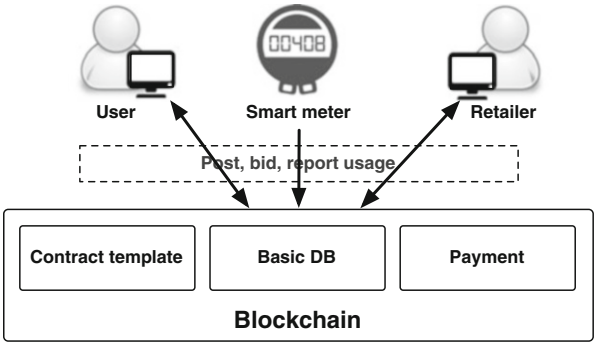

Fig. 4.10 Architecture of blockchain-based electricity contracts using smart meters. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

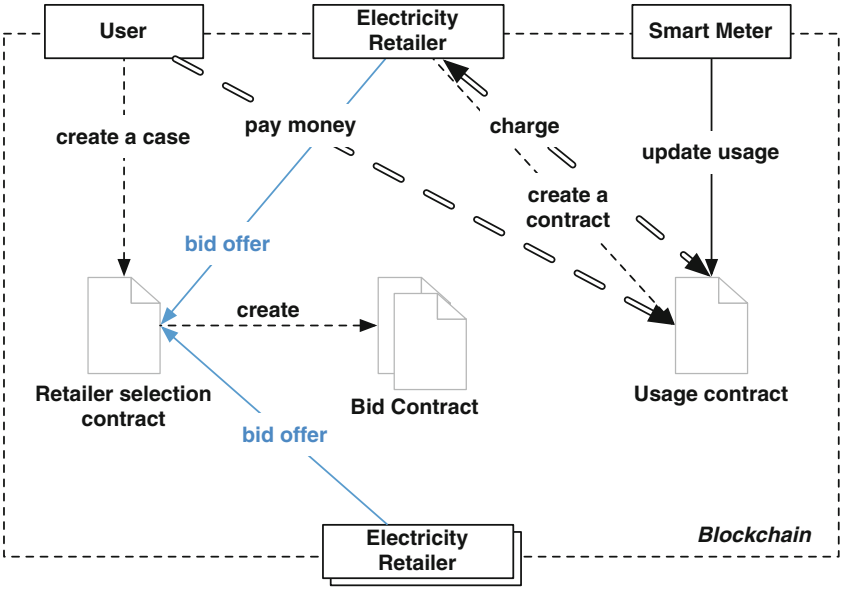

Fig. 4.11 Interaction of smart contracts among themselves and with other entities. © 2017 by the Commonwealth Scientific and Industrial Research Organisation, reprinted with permission

as they prepare their offer. The user's retailer selection contract collates all the bids, which can then be shown on a web page accessible to the user for final selection. The interaction with and among the relevant smart contracts is shown in Fig. 4.11.

After the electricity retailer is selected, a usage contract that is specific to the pair of the user and the retailer is generated and uploaded to the blockchain. The usage contract is used by the smart meter to report the monthly usage. There are two options to pay the bills. The user could either pay the bill to the retailer's

account directly before the deadline or deposit money into the contract and let the retailer withdraw from it when the payment is due. When the user decides to switch retailers, she could create a replacement usage contract from the new retailer selection contract. The usage contract with the previous retailer will terminate after the new contract is created.

4.4.4 Non-functional Property Discussion

Integrity The conventional approach relies on individual electricity retailers to maintain the internal system and usage data. New usage data is validated solely by the electricity retailer. In the blockchain-based design, usage records can only be created by the smart meters using their private keys. All transactions are validated by all processing nodes in the blockchain network. Electricity retailers hold local copies of the blockchain, through which they can access the historical electricity usage of any smart meter.

Privacy In the conventional design, the usage data is only shared with the current electricity provider. In contrast, in the blockchain-based design, the data is shared with all electricity providers that are on the consortium blockchain. Design of the blockchain-based system is important to meet privacy requirements. For example, the blockchain might be a private permissioned blockchain or distributed ledger, and users and smart meters might access the blockchain only through controlled web interfaces or APIs.

Transparency In the conventional design, information from smart meters and historical usage are stored by each separate electricity retailer. Such information is not accessible by other electricity retailers. The consortium blockchain used in the blockchain-based design provides a common shared data storage for historical usage associated with any electricity retailer.

4.5 Further Reading

This chapter is partly based on our earlier works (Staples et al. 2017).

A detailed use case from a startup company, focussing on the reduction of counterparty risks in agricultural supply chains, is described in Chapter 12.

A model-driven approach is proposed in Weber et al. (2016), which can generate smart contracts automatically from process models. This approach is also discussed in Chapter 8.

Reports from the World Bank (Ratha et al. 2016; The World Bank 2016) provide data and insights about remittance.

Bitcoin has been applied to smart meters deployed in South Africa (Prisco 2015), where each smart meter is equipped with its own Bitcoin address. The smart meters

directly pay for their metered electricity and water supply from their balance and form an interesting middle ground between prepaid and post-paid services. Bitcoin has also been applied in smart grid scenarios (Dimitriou and Karame 2013) to facilitate aggregating energy production and consumption reports without relying on a single point of trust. This enables anonymous tasking and privacy-preserving billing and barter of energy.

The use of smart contracts to enable machine-to-machine communication in IoT has, for example, been demonstrated by the ADEPT (Autonomous Decentralized Peer-To-Peer Telemetry) project (IBM 2015).