Digital Tokens

Another common conversation point in discussions around blockchain is tokens, or digital tokens to be more specific. A digital token can be any kind of digital asset or any digital representation of a physical asset. Within Ethereum a digital token can represent any fungible, tradable good, such as currency, reward points, gift certificates, and so on. All Ethereum-based tokens implement baseline functionality in a standard way, known as the ERC20 token standard. These tokens remain compatible with any other client or contract that relies on and uses the same standards.

The significance of digital tokens is that in recent years it was venture capital companies and institutional investors that had invested in digital currency startups, to the tune of approximately $1 billion. This was one of the typical ways that a startup raised money. Traditionally, companies raise money in three ways:

- **Equity** The company sells shares (ownership) in exchange for funds.
- **Bond** The company borrows cash and promises to repay the amount with interest.
- **Presell product or service** The company takes orders and payment for a product or service that does not exist, isn't complete, or is not available yet, such as event tickets, mobile or console games, or Kickstarter projects.

With these traditional approaches there is a greater degree of government regulation put in place to prevent investors from being scammed by swindlers and crooks.

What was happening in the funding aspect of these digital currency startup efforts? A new method, called an initial coin offering (derived from IPO, initial public offering), provided another way for a company or project to raise capital. These new projects raised money by creating and then selling their own digital tokens through crowdfunding on a blockchain. Some interesting aspects to this approach are that the tokens are the currency that is used in the company/application that is being created or developed. The developers of the application (or contributors) are paid in these tokens. The tokens can be converted to any currency at the prevailing market rate when needed. This really provides the ability to create and launch a project using a decentralized business model (DBM). There is a significant difference and divergence between these decentralized approaches to raising money and the traditional centralized ways. In effect, these ICO/DBM projects create their own financial ecosystem in order to drive the development, innovation, and delivery of the project.

This model is decentralized in several ways. There is no central control structure gatekeeping or preventing participation; it has shared contributions and ownership by all participants. This business model is facilitated by platforms that are already in place—the Internet and cryptocurrencies. This really is being tried for the first time and is looking like the future model of blockchain and the applications that will be created to run on it. It is nonetheless very important that these tokens are structured in a way that prevents them from coming under regulatory scrutiny. The main tests that regulators apply when scrutinizing these arrangements come from the U.S. Supreme Court case *Securities and Exchange Commission (SEC) v. W. J. Howey Co*. This 1946 case established the test for whether an arrangement involved an "investment contract." The case concerned an offer of a land sales and service

contract that the court upheld to be an investment contract (in other words, a security) within the meaning of the Securities Act of 1933. In terms of tokens on the blockchain, the following three items must be true for a digital token to be considered an investment contract (security):

1. An investment of money
2. In a common enterprise or business
3. An expectation of profits predominantly from the work of others

A full and detailed document is available at https://www.coinbase.com/legal/securities-law-framework.pdf that attempts to lay out a process and best practices to help application builders—that is, those who want to create and sell tokens to the public—avoid such tokens being classified as a financial security.

Financial Services Use Cases

Blockchain will continue to transform the financial services industry because of the benefits and features it can provide, some of these being faster throughput, reduced costs, less room for error, transparency, and a multitude of "ities"—quality, reliability, simplicity, and traceability. The financial institutions that find ways to adopt and apply blockchain technology will gain the competitive advantages of delivering solutions with a faster time-to-market at a reduced cost. We will now examine use cases that will show you how blockchain can be applied to real-world challenges in the financial services industry.

Know Your Customer (KYC) Use Case

Problem: Know Your Customer (KYC) is the process by which a financial institution gathers information about a customer. The main purpose of this process is to ensure that institutions' services are not misused, and this process takes place when a customer opens an account. The process varies because each financial institution is responsible for being in compliance with the requirements laid out and specified by their local regulatory body. The process typically requires the passing of documents back and forth between the customer and financial institution. There is very little automation; financial institutions dedicate a huge amount of resources to the process, but it is still very time consuming. These delays are frustrating for a customer who wants to use the institution's services immediately. So how does blockchain technology help here? See Figure 2-2 for a blockchain solution.

Solution: The customer's personal information, KYC documentation, and data are encrypted and added as a block in the blockchain. The customer's block is validated using a consensus model running on the network. When the customer wants to open an account with a bank or financial institution (FI) for the first time, the FI directs them to their block and authorizes them with access. The FI can access all of the validated KYC information and move the customer along the on-boarding process. The single source of KYC (identifying) data that can easily be shared between financial institutions and external agencies will eventually result in much faster account opening, reduced resources, and therefore lower costs. This can all be done while maintaining the privacy of data, because the owner of the data (the customer) strictly controls its access. Initially, there would still be a role for a third party (e.g., the initial bank),

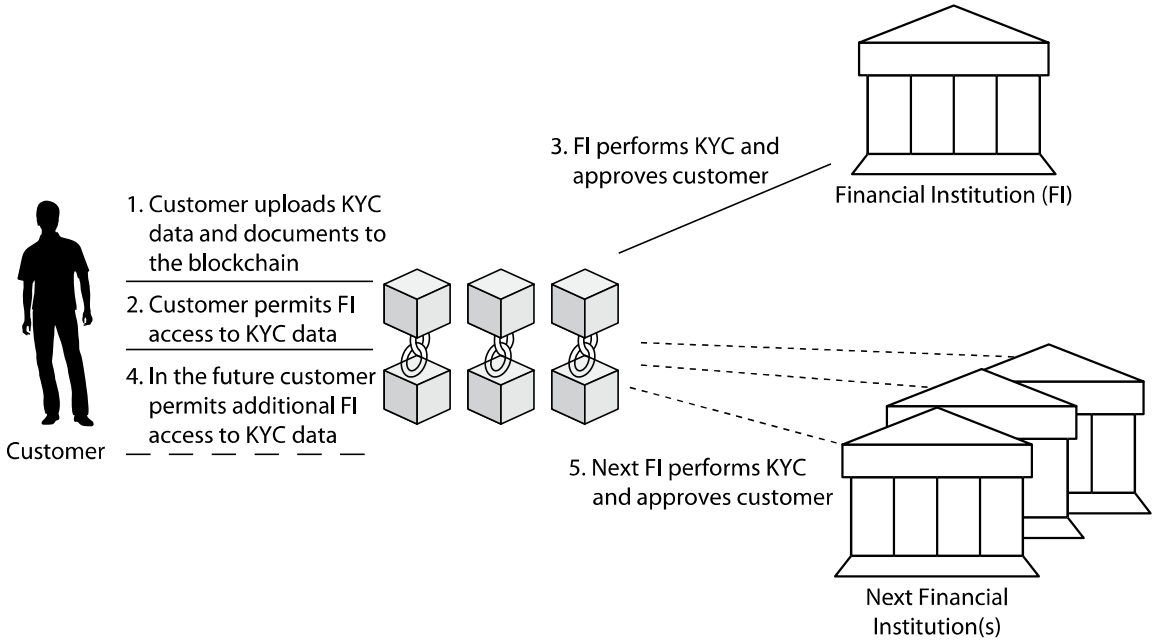


FIGURE 2-2 **Know Your Customer (KYC) use case solution**

and they might be rewarded for doing the initial physical work of adding information to the blockchain. This would happen at least until a decentralized attestation system was created as the process evolved.

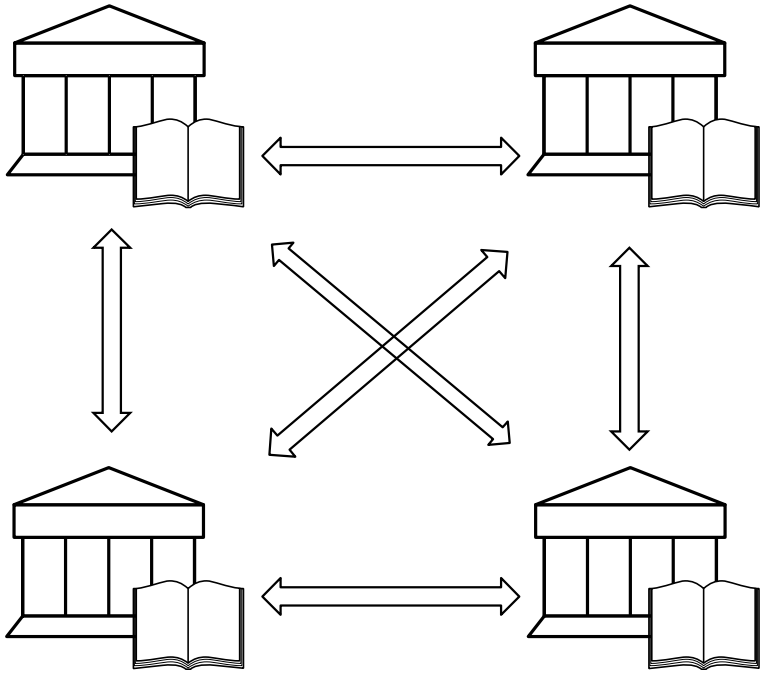
Asset Management Settlement Use Case

Problem: Typical back-office trade processing and settlement methods can be awkward, risky, and time consuming. This is especially the case when manual steps are involved, such as during matching and reconciliation tasks. Every entity in the value chain keeps its own copy, or view, of the transactions that have taken place. These processes continue to get more complicated as new instruments are being developed and traded. In short, mistakes and associated costs are rising in proportion to the complexity of the tasks and involvement of resources to deal with them. See Figure 2-3 for a solution to asset management settlement.

Solution: The transactions can be stored in a blockchain and made available to all parties. This would drastically simplify and streamline this entire process for all parties. It would lead to automation of the trade life cycle, facilitate easier management of the data, and provide transparency as well as substantial infrastructure and resource reductions. These improvements would lead to minimal reconciliation and ultimately faster processing times. While all of the transactions may be visible to everyone, it is possible to arrange and set up so that only certain parties are privy to this information.

Insurance Claims Processing Use Case

Problem: Have you ever needed to make a claim on an insurance policy? If you have, you'll be able to relate to this problem. The claims process is complex and often drawn out and lengthy. Insurance contracts are typically difficult to understand, because over time they have evolved into a complex web of legal language in reaction to prior unfavorable outcomes. So when



A blockchain solution will remove the need for intermediaries and provides a trusted and shared (with permission) view of data:

- Increase availability—no downtime
- Reduce costs because there are fewer reconciliation issues
- Speed up settlement because validation is fast
- Improve transparency and ability to monitor

FIGURE 2-3 **Asset management settlement use case solution**

you need to make a claim, it takes a while for the process to complete. Insurance companies are composed of many separate department silos, such as underwriting, policy issue and administration, claims, actuarial and statistical, accounting, investment, legal, and audit. When you add manual processes, legacy models, and disjointed data elements to the daily onslaught of fraudulent claims, it stands to reason that "speedy" is not the word typically associated with customer outcomes. See Figure 2-4 for a solution to insurance claims processing.

Solution: Create insurance policies using a smart contract on the blockchain. The qualities of a smart contract (control, transparency, and traceability) would allow for much more automation. A smart contract would provide the customer and insurer with the ability to manage claims in an open, speedy, and indisputable way. The contract (policy) is uploaded to the blockchain and validated by the network. Similarly, claims are then uploaded to the blockchain and applied to the smart contract. That being said, blockchains cannot access data outside their network. This data can represent external conditions such as temperature, payment, price change, or RFID presence trigger. An oracle (or data provider) is a third-party service designed for use by smart contracts. An oracle provides the necessary external data and pushes it onto the blockchain. Then the contract and the network will be able to validate and enforce the claim and either automatically reject or accept it. When the correct conditions are met, a payment is automatically triggered.

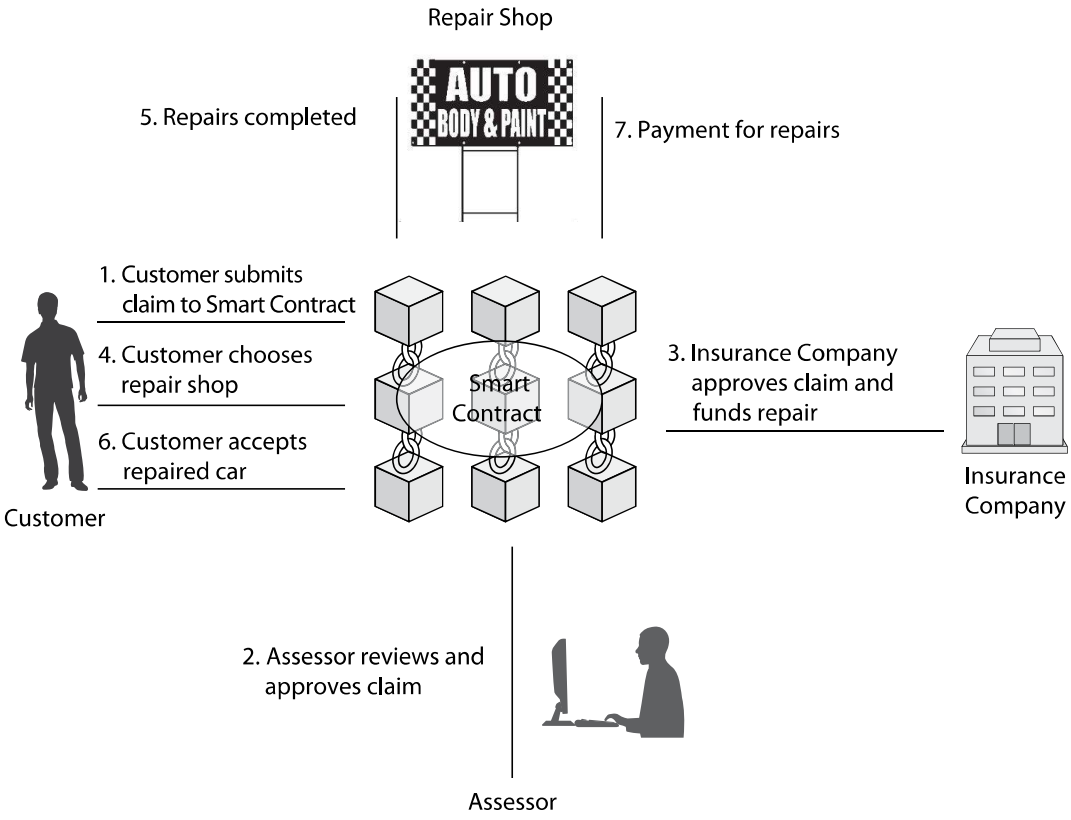


FIGURE 2-4 **Insurance claims processing use case solution**

Trade Finance (Supply Chain) Use Case

Problem: *Trade finance* refers to financial transactions, both domestic and international, that relate to trade receivables finance and global trade. Trade finance is a core business function for all global banks, especially tier 1 banks. Given its importance, it still lags in its application of technology and still resorts to using very manual processes for its document-centric flows. This leads to interruption in business cycles, and the lack of transparency leaves the door open to financial crime. Supply chains between many parties are complicated, distributed, and lack trust, therefore they are very slow and need many third parties such as banks and clearinghouses to facilitate the trust aspect and allow the commerce supply chain to flow. See Figure 2-5 for a solution to supply chain finance.

Solution: Blockchain will hold all of the necessary information in a smart contract, updated instantly and viewable by all members on the network. The smart contract can be used to automate the transfer of title to goods and money. This automation and network validation remove the need for third-party facilities, such as letters of credit (LCs), and will help to streamline the whole process and measurably reduce costs by eliminating the third parties and their associated fees. Applying smart contracts results in:

- Faster cycle time. Reducing and eliminating human intervention yields a more efficient process, resulting in much shorter cycles.
- Reduced fraud. The open and auditable transactions available in the distributed ledger reduce or eliminate supplier fraud.

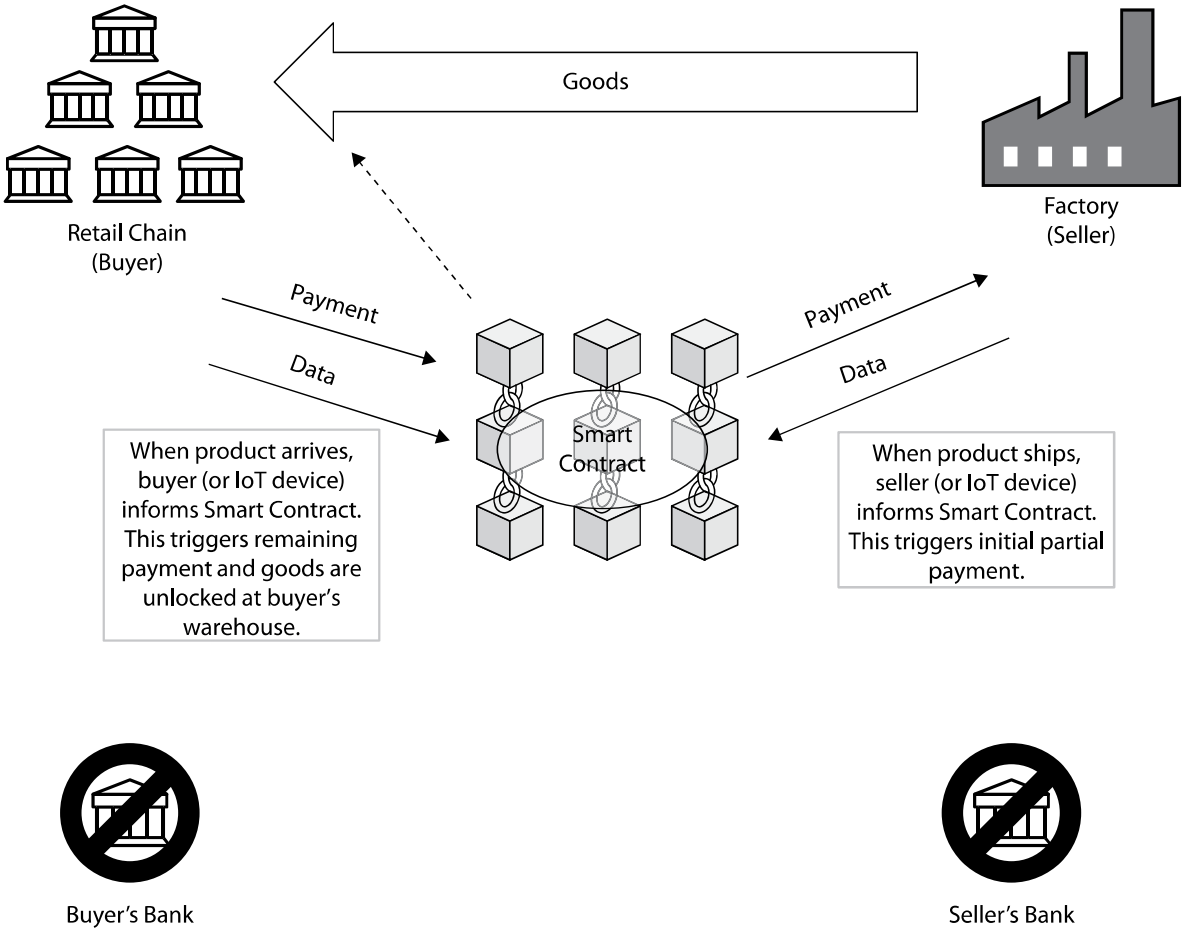


FIGURE 2-5 **Trade finance (supply chain) use case solution**

- Reduced costs and fees. The shared ledger is available to all banks, rating agencies, and suppliers participating in the chain, reducing processing and storage costs and fees. Payments between institutions in the network can potentially be made through a cryptocurrency. This will eliminate the need for higher-cost third-party payment networks.

Global Payments Use Case

Problem: It can take days to transfer money from a party in one country to a party in another country. The global payments business is a large, slow, costly, and error-prone industry. It is also attractive to those who wish to engage in money laundering because it is not completely traceable. See Figure 2-6 for a solution for cross-border payments.

Solution: Blockchain payment technologies can add tremendous value in this space by (a) reducing the multi-day payment cycle down to real time, (b) enhancing currency conversions, and (c) providing transparency to improve anti–money laundering capabilities. Aside from Bitcoin, there are several other cryptocurrency solutions that are gaining popularity in the

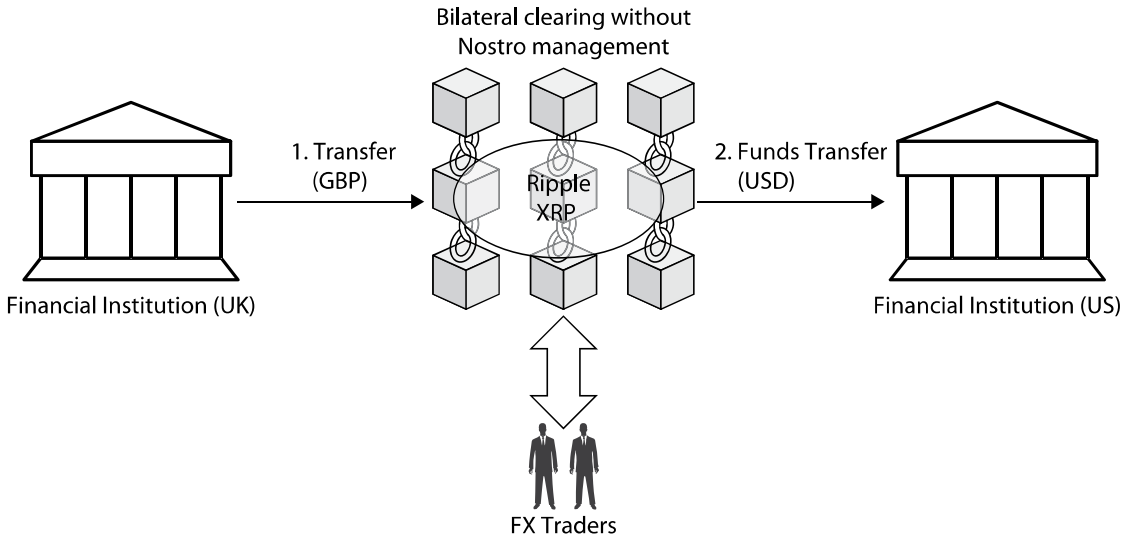


FIGURE 2-6 **Global payments use case solution**

payments space, notably the Ripple network and Ripple XRP. Santander became one of the first banks to apply blockchain to payments, enabling customers using their application to make overseas payments that clear within 24 hours. Blockchain will eventually enable real-time payments while reducing operational costs, human error, and fraud.

Smart Property

Smart property is really the extension of smart contracts reaching out into the practical and interactive world that includes the Internet of Things (IoT). We'll cover smart contracts later in this chapter, but feel free to to jump ahead and read about them first and come back to this section.

We've all seen (or, if not, certainly read about) actual physical objects being connected to the Internet. For example, with home automation the smart home has smart light bulbs, smart locks, and a smart refrigerator and a smart oven. All of these items are called smart because they have computer technology embedded within, they are programmable in some way, and they are connected to the Internet. So, in the case of a light bulb, it can be customized, scheduled, and controlled remotely. See Figure 2-7 for additional examples of smart property.

Smart property is all about ownership, access, and control of things using the blockchain network. Smart property can be in the physical world, such as a vehicle, tablet, or even real estate. In the virtual (non-physical) world, smart property can be a financial instrument, trademark, copyright, or patent. The real advantage to making property smart is that it can be traded, accessed, and controlled in a near trustless way, reducing cost as well as fraud. This will open up and expand commerce, making everything more competitive and at the same time less expensive. Think about this: smart property can be used as collateral by a borrower when they borrow money from a lender using a smart contract. We envision the blockchain becoming a tool for inventory, tracking, and exchange of assets in the future.

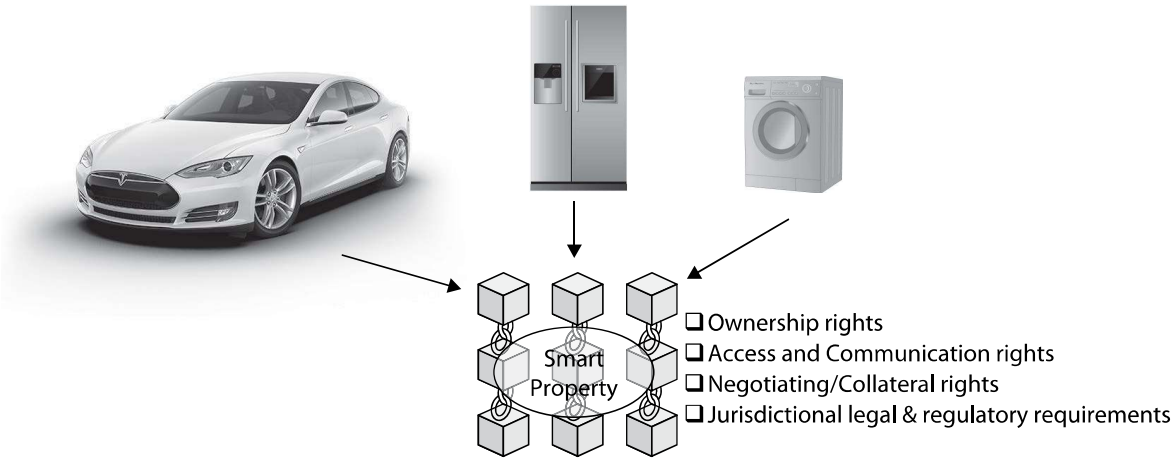


FIGURE 2-7 Smart property

Transferring Ownership of Smart Property

While simple forms of smart property exist, such as the combination of your car ignition and the car key, more advanced forms will be needed in the future to allow transfer of ownership. We will now distill a smart property approach to the ownership of a car. When the manufacturer builds the car, the car is given a digital certificate signed by the manufacturer and an identification key containing the public part of the certificate. The car has built in enough technology so that it can provide valid proof of its existence, and other valuable data like ownership, build date, maintenance history, and distance traveled or hours used. The car would also contain a small amount of cryptocoins (*C*) deposited on the ownership key to facilitate transfer in the future. See Figure 2-8 for an example flow for transferring ownership of smart property.

Now let's look at the steps involved when the car gets sold:

1. The buyer generates a random number (nonce) and passes it to the seller and asks in return for the car data.
2. The seller takes the buyer's random number and passes it to the car (probably implemented via a touchscreen or an application on the seller's phone that is paired to the car already).
3. The car returns a signed (via identification key) data structure that contains the buyer's random number, the car's public key, any pertinent data (mileage, age, etc.), the current owner's public key, and the transaction details of the prior ownership transaction. This data structure provides sufficient information for the buyer to know what they are buying and who the seller is.
4. The seller identifies the key to receive payment (*SellerKey*) and price required (*P*).
5. The buyer generates a new ownership key (*BuyerKey*) and creates a skeleton transfer transaction containing a pair of inputs and a pair of outputs. Note that the transaction is not completely valid and cannot be fully executed because only Input#1 is signed:
   - Input#1 is a signed entry for P cryptocoins.
   - Input#2 identifies the ownership key (holding the C cryptocoins).

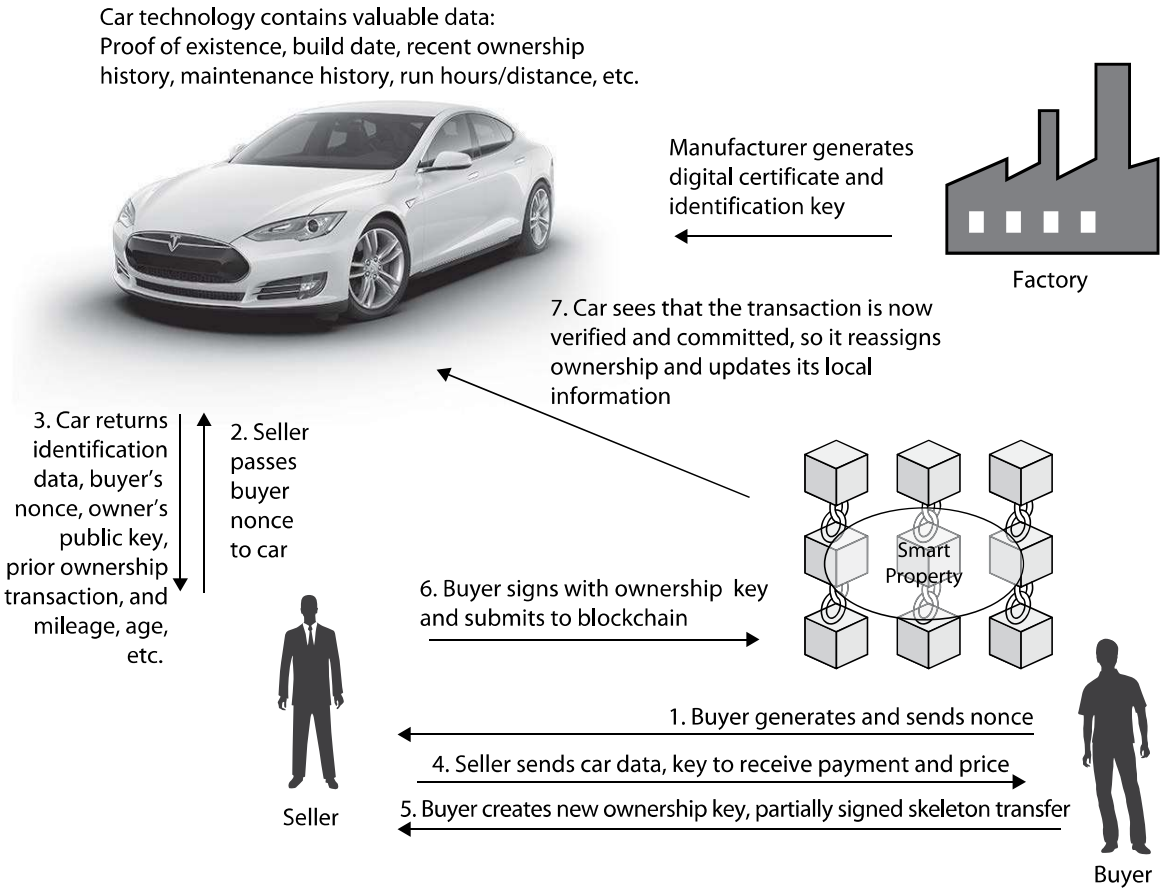


FIGURE 2-8 Transferring ownership

- Output#1 is an instruction to send P cryptocoins to the SellerKey.
- Output#2 is an instruction to send C cryptocoins to the BuyerKey.

6. The buyer passes this partially signed transaction back to the seller. The seller signs Input#2 with the car's current ownership key, commits the transaction to the blockchain, and waits for it to be validated.
7. The buyer presents the car with the verified blockchain transaction. This is a Merkle branch hash of the block header of the transaction and enough additional block headers to fill in the gap from the car's current ownership transaction. The car sees that the new transaction reassigns ownership and updates its ownership information.
8. The buyer now owns the car.

As we mentioned earlier, there will most likely be a touchscreen in the smart car and an application on the buyer's and seller's phones to facilitate identification and the transfer. It is highly likely that our smartphones become even more integral with our lives and even become part of the keys to our smart property along with some biometric or additional authentication/authorization methods.

Using Smart Property as Collateral

While transferring ownership of smart property is an obvious necessity, using it in other ways will also be possible. For example, what if you could leverage your smart property and use it as collateral in another contract? This isn't transferring ownership, although penalties for not fulfilling a contract may lead to transferring ownership of the smart property, but while it's being used as collateral you will most likely want to continue to use the smart property. What would be the sense in metaphorically sticking it in a drawer where it would do nothing for the life of the contract? So let's flesh this out a little with an example. You want to borrow money. In exchange, the people lending you the money want something pledged as security for repayment of the loan, to be forfeited in the event of a default. You put up your smart property as collateral, but the wrinkle is that you still want to use the smart property while the loan is in place. See Figure 2-9 for an example flow for using smart property as collateral for a loan.

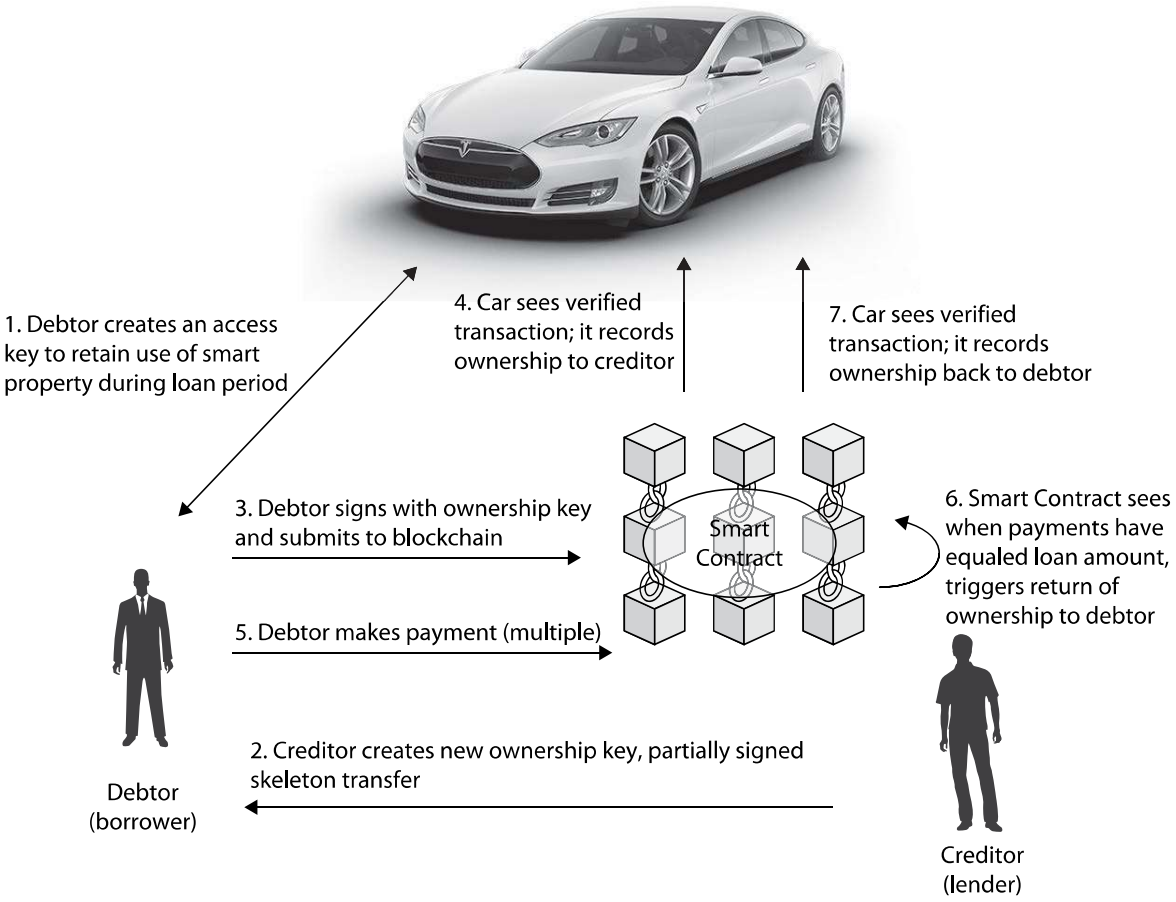


FIGURE 2-9 Using smart property as collateral

So how will we do this in practice? As we said earlier, smart property can manage ownership, access, and control. These are the basic capabilities. In this scenario we will use the ability to program into the contract the state of ownership and access for the life of the loan. Here is a high-level digest of how a loan contract using smart property as collateral might be set up:

1. The debtor creates an access key for the smart property so they can keep using it for the duration of the loan provided that they maintain the payments.
2. The creditor generates a key (*CKey*) that is to receive payments to repay the loan amount (*LAmt*).
3. The creditor generates a new ownership key (*OKey*) and creates a skeleton transfer transaction (*Tran*) containing three inputs and two input/outputs. The creditor passes this transaction to the debtor. Note that the transaction is not completely valid and cannot be fully executed because only two of the inputs are signed:
   - Input#1 is a signed entry for LAmt (loan amount).
   - Input#2 identifies the repayment amount (initially zero).
   - Input#3 identifies the current ownership key (smart property).
   - Input/Output#1 identifies the DKey and an instruction to send LAmt to DKey and assign ownership of smart property to OKey.
   - Input/Output#2 signs entry OKey and an instruction to assign ownership to DKey when loan is repaid.
4. The debtor signs Input#3 with the ownership key and Input/Output#1 with the DKey.
5. The debtor makes payments to the loan by adding amounts to Input#2.
6. When the transaction Input#2 reaches the value in Input#1, the instruction to take back ownership of the smart property is executed.

There are some additional conditions that would be attached to this contract, namely time limits. The creditor can revoke the access token if the loan is not repaid by a certain time, because they have the ownership key. The smart property can then be taken and used by the creditor or sold to recoup the loan amount. Now let's take a look at the smart contract itself.

Smart Contracts on the Blockchain

Blockchain technology handles cryptocurrencies and tokens as mentioned earlier in this chapter and throughout the book; it is the underlying platform for a new way of organizing and managing relationships. These include legal relationships and contracts. In order to best explain what a smart contract is, we must first look into the problem of trust.

The Trust Problem

The trust problem has been around since the dawn of time. In order to progress, a society or group of people has to cooperate with each other. When people cooperate they can do more collectively that they could individually. However, in doing so they are also opening themselves up to being deceived, misled, and subsequently disappointed. To attempt to address this issue,

societies have instituted rituals, passed laws, and even installed governance processes. All of these elaborate techniques are in place to address the trust problem. To better explain the trust problem, let us use an example. A man frequently uses his credit card to pay for goods and services. He can walk into a store, pick an item, go to the checkout, and pay with the credit card. The store allows the purchase because the credit card, when swiped or inserted into the chip reader, checks with the bank to see if the shopper is a good risk for them to authorize the purchase and confirm that he is the person to whom the card was issued. The bank will actually collect the payment from the shopper some time later when it bills him. So, you see, there is trust all around here. The store trusts the bank and the shopper, and the bank trusts the store and the shopper, and the shopper trusts the store and the bank. The shopper can carry around a small piece of plastic instead of a wad of cash. If he loses the card, he can have it replaced within a few days. If he had to carry the cash equivalent, he might lose it or get robbed and would never get it back. Having the trust element makes it easier to purchase items, and that's also good for merchants, who always want and need to sell more items. As the barriers to payment come down, a whole lot more commerce occurs. That brings us back to the original point that trust is needed for cooperation, which in turn leads to progress.

Trusted Third Party

One way to solve the trust problem is to use a trusted third party. The bank and credit card example given above is this exactly. The transactions between the customer, the merchant, and the shopper are passed through and logged by a bank (credit card issuer). The bank facilitates the transaction; see Figure 2-10 for a diagram depicting a trusted third party. The bank can also step in and resolve a dispute in the event that a customer finds a transaction on their monthly statement that they didn't make.

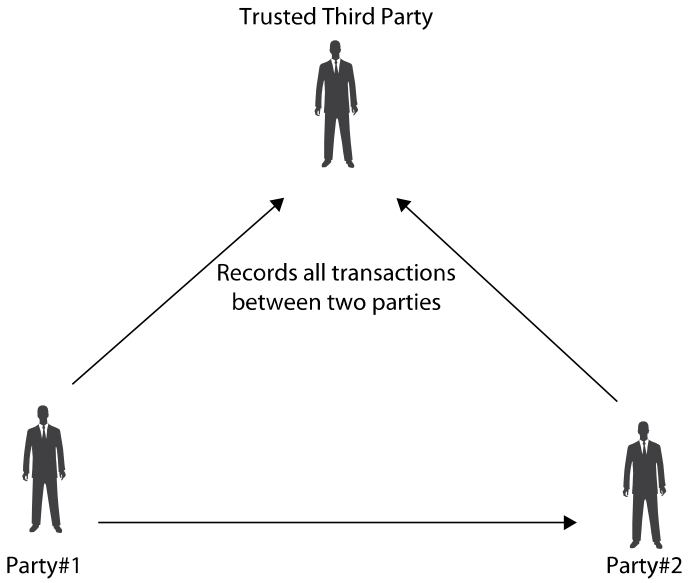


FIGURE 2-10 Trusted third party

Distributed Ledger and Consensus

What we have just described is still a centralized way to solve the trust problem. There is another way to solve it using a distributed ledger (or shared ledger) and combining this with a consensus methodology. Figure 2-11 depicts a ledger that is distributed across multiple parties. So, instead of logging transactions with a single third party, you send a single copy of each transaction to all parties in the network. All parties in the network would be required to keep an ongoing ledger of all transactions. Therefore, every party in the network would have the exact same set of transactions. At a point in time, everybody would know how much the shopper owes the merchant. If a dispute arises, the consensus majority (51 percent) of the network of ledger keepers would decide what he owed.

Blockchain technology follows this distributed ledger and consensus method. It is a network for resolving the trust problem through a distributed (decentralized) and publicly verifiable (open) ledger.

Blockchain Details

Blockchain leverages and combines the networking capabilities of computers with cryptographic technology to store and process data. Any computer on the network (known as a node) can be located anywhere with Internet connectivity. It's really outsourcing of intelligent computing resources to the cloud. These resources provide a platform for developers to build applications. It can be likened to other Platform as a Service (PaaS) offerings that exist today. The blockchain platform logs, processes, saves, and verifies transactions. Because it is based on the shared ledger concept, each and every node on the network stores the same copy of the ledger and therefore all of the transactions taking place on the platform. It is this decentralized (shared) aspect of the ledger that makes blockchain the public, comprehensive, permanent, and verifiable authority for administering and storing records of transactions. Let's look at an example of how this shared ledger works.

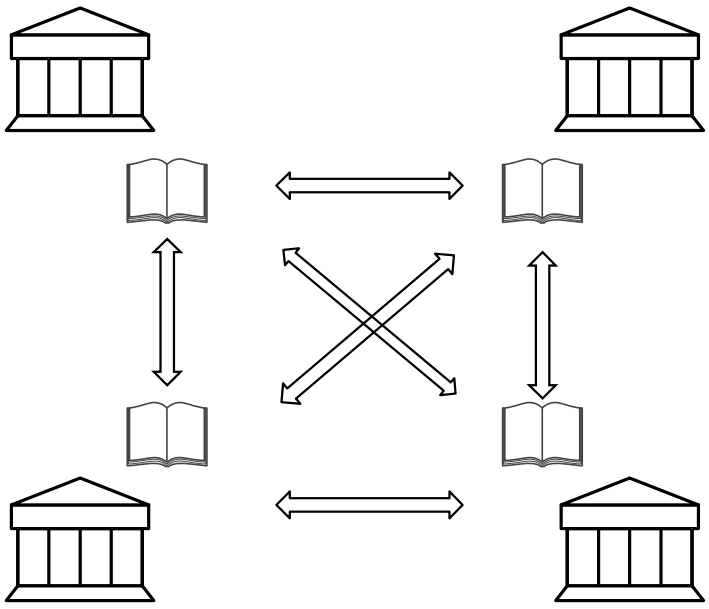


FIGURE 2-11 Distributed ledger and consensus

In this scenario, there are two parties who want to exchange money, with Party#1 wanting to give 100 cryptocoins to Party#2. As in the real world, each party will need a place to hold their cryptocoins—a digital wallet. Provided Party#1 knows the address of Party#2, they will be able to create a transaction that transfers the 100 cryptocoins to Party#2. This transaction is added to a temporary block of data that is then stored on the network. This block contains all of the transactions that took place on the network during a particular length of time, for example, the last few minutes. While it is recorded on the network, this block has not been verified. Once a set interval of time passes, all of the data collected in the temporary block is gathered and the network holds a competition among the participants on the network. The winner of the competition takes on the task of processing and verifying the transaction on the block and committing it to the permanent record by adding the block to the blockchain. Once they do this, the winner broadcasts the proof of work to the whole network; each node then checks that the proof of work is correct. If 51 percent of the nodes agree that the proof is correct, the block is added as a permanent block to the blockchain.

While the blockchain is often referred to as a trustless system, what that really means is that there is no need to place any trust in any human interaction because everything is taken care of by the platform technology. The only interesting point here is that the participants on the network give up their computing resources to this process. What is their incentive to do so? The winning participant is rewarded with a digital token for the underlying platform—that is, the winner gets a newly minted cryptocoin. So, are you ready to commit your home computer to the blockchain network and become a miner? Think again: to give yourself a shot at winning the competition requires a large amount of computing resources. Some of the largest companies or entities that participate are based in China, where electricity is state subsidized. This winner-takes-all approach could have been done in a more equal way where all participants could have shared the reward, but perhaps the competition is what made the platform so successful. Perhaps a future platform will take another look at the reward system to see if another way will work.

Smart Contract

In the real world, a contract is an agreement between parties that is intended to be enforceable by law. Within these contracts are agreements that lay out what each party is to do to fulfill the contract. In a two-party contract, the essence of this can be likened to an Aristotelian syllogism, which is a fancy way of saying "if this is true, then the following will happen, else the following will happen." If one party fails to carry out their obligation, they can be taken to court and ordered to carry it out or to provide financial compensation to the other party. In computer programming terms, these Aristotelian syllogisms are implemented using an "if … then … else …" construct. A smart contract is one or more of these conditions combined with the capabilities to enforce the obligations automatically. The distributed ledger system contains all of the data and theoretically the capabilities necessary for the smart contract to execute autonomously. The required conditions are coded in the smart contract and once they are met the contract obligations are automatically executed. In this digital case, though, there is no need for the courts or mediation—the facts are available to the contract, so it cannot make the wrong decision.

Example Smart Contract

The following is the sample code that Ethereum recommends will represent a digital token in their ecosystem. A token can represent any fungible and tradable good. This can be coins, reward for customer loyalty, certificates, and so on. Because digital tokens implement some basic data and functionality in a standard way, they will operate in a consistent way. If you use this code, the token, because it is ERC20 compliant, will be instantly compatible with the Ethereum wallet and any other client or contract that uses the same standards.

LISTING 2-1 Ethereum Smart Contract Token (ERC20) Example (*Continued*)

```
pragma solidity ^0.4.8;

contract tokenRecipient { function receiveApproval(address _from, uint256 _value, address _token,
bytes _extraData); }

contract MyToken {
    /* Public variables of the token */
    string public standard = 'Token 0.1';
    string public name;
    string public symbol;
    uint8 public decimals;
    uint256 public totalSupply;

    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;
    mapping (address => mapping (address => uint256)) public allowance;

    /* This generates a public event on the blockchain that will notify clients */
    event Transfer(address indexed from, address indexed to, uint256 value);

    /* This notifies clients about the amount burned */
    event Burn(address indexed from, uint256 value);

    /* Initializes contract with initial supply tokens to the creator of the contract */
    function MyToken(
        uint256 initialSupply,
        string tokenName,
        uint8 decimalUnits,
        string tokenSymbol
        ) {
        balanceOf[msg.sender] = initialSupply;              // Give the creator all initial tokens
        totalSupply = initialSupply;                        // Update total supply
        name = tokenName;                                   // Set the name for display purposes
        symbol = tokenSymbol;                               // Set the symbol for display purposes
        decimals = decimalUnits;                            // Amount of decimals for display purposes
    }
```

LISTING 2-1 Ethereum Smart Contract Token (ERC20) Example (*Continued*)

```
/* Send coins */
function transfer(address _to, uint256 _value) {
    if (_to == 0x0) throw;                                  // Prevent transfer to 0x0 address.
                                                            // Use burn() instead
    if (balanceOf[msg.sender] < _value) throw;              // Check if the sender has enough
    if (balanceOf[_to] + _value < balanceOf[_to]) throw; // Check for overflows
    balanceOf[msg.sender] -= _value;                        // Subtract from the sender
    balanceOf[_to] += _value;                               // Add the same to the recipient
    Transfer(msg.sender, _to, _value);                      // Notify anyone listening that
                                                            // this transfer took place
}

/* Allow another contract to spend some tokens in your behalf */
function approve(address _spender, uint256 _value)
    returns (bool success) {
    allowance[msg.sender][_spender] = _value;
    return true;
}

/* Approve and then communicate the approved contract in a single tx */
function approveAndCall(address _spender, uint256 _value, bytes _extraData)
    returns (bool success) {
    tokenRecipient spender = tokenRecipient(_spender);
    if (approve(_spender, _value)) {
        spender.receiveApproval(msg.sender, _value, this, _extraData);
        return true;
    }
}

/* A contract attempts to get the coins */
function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {
    if (_to == 0x0) throw;                                  // Prevent transfer to 0x0 address.
                                                            // Use burn() instead
    if (balanceOf[_from] < _value) throw;                   // Check if the sender has enough
    if (balanceOf[_to] + _value < balanceOf[_to]) throw;  // Check for overflows
    if (_value > allowance[_from][msg.sender]) throw;     // Check allowance
    balanceOf[_from] -= _value;                             // Subtract from the sender
    balanceOf[_to] += _value;                               // Add the same to the recipient
    allowance[_from][msg.sender] -= _value;
    Transfer(_from, _to, _value);
    return true;

}

function burn(uint256 _value) returns (bool success) {
    if (balanceOf[msg.sender] < _value) throw;              // Check if the sender has enough
    balanceOf[msg.sender] -= _value;                        // Subtract from the sender
```

LISTING 2-1 Ethereum Smart Contract Token (ERC20) Example

```
        totalSupply -= _value;                                  // Updates totalSupply
        Burn(msg.sender, _value);
        return true;
    }

    function burnFrom(address _from, uint256 _value) returns (bool success) {
        if (balanceOf[_from] < _value) throw;                // Check if the sender has enough
        if (_value > allowance[_from][msg.sender]) throw;    // Check allowance
        balanceOf[_from] -= _value;                          // Subtract from the sender
        totalSupply -= _value;                               // Updates totalSupply
        Burn(_from, _value);
        return true;
    }
}
```

Blockchain IoT Protocol Projects

The Internet of Things (IoT) is a major buzzword, and there is little doubt that it is an industry with tremendous growth in its future. According to Gartner, the number of connected devices will exceed 20 billion by 2020, with a market worth of more than $3 trillion. A connected device can be as simple as a small sensor (think motion detector) or scale up to a much more complicated appliance (think house, car, boat, airplane), as depicted in Figure 2-12. One of the biggest concerns with these connected devices is security, and this breaks down into two aspects. The first is preventing unauthorized access to the device itself, and the second is unauthorized access to any data being exchanged with the device.

Additionally, as with all other data processing tasks, you will eventually run into scalability issues with a centralized approach, especially with the billions of transactions that are expected on these devices. Centralized servers can also be vulnerable to single point of failure conditions, making these devices susceptible to denial-of-service attacks where servers are flooded with traffic from compromised devices spread across the world. This is exactly how IoT systems that handle sensitive tasks will be impacted if they are not architected with resiliency from the start. That's where blockchain technology is being offered as a solution to the challenges of IoT. Each device would continue to manage and administer its own behavior, security roles, and the rules for interacting with other devices. Blockchain would be the platform to facilitate the transactions and the coordination between these devices. Blockchain technology will enable secure mesh networks to be created. The devices on the network can interconnect in a reliable way and prevent security threats that use techniques like spoofing and/or impersonation. If each valid point is registered on the blockchain, this will facilitate identification and authentication in a decentralized way, and the network will be scalable to support billions of anticipated devices.

In early 2017, a group of industry-leading startups and companies came together at a summit to discuss the challenges facing blockchain and IoT innovation and the potential for a collective effort to address them. This summit was the first step to explore and build a shared blockchain-based Internet of Things protocol. There were presentations that exchanged

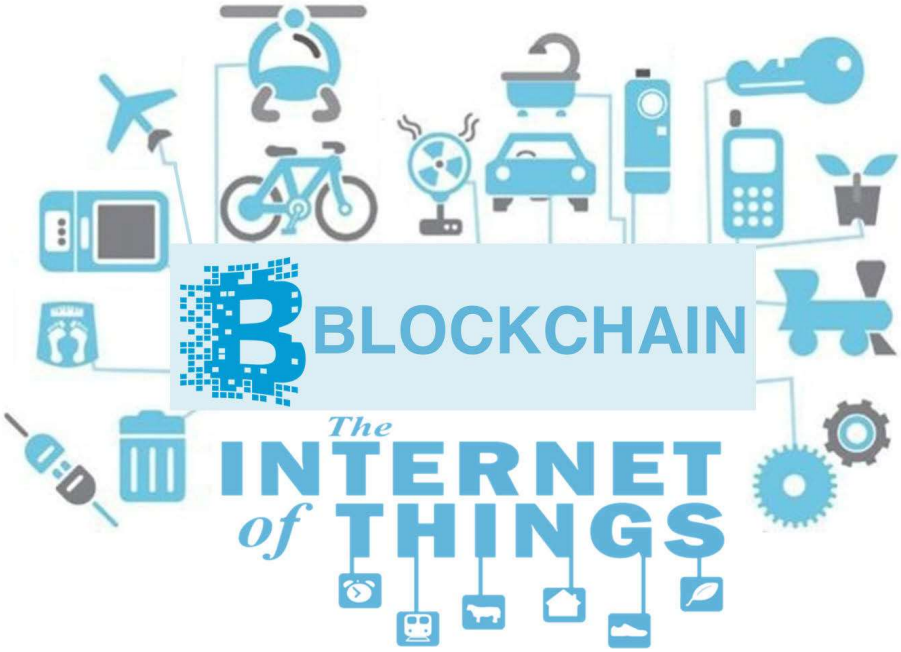


FIGURE 2-12 Graphic depiction of IoT examples

ideas on applicable use cases, findings and feedback within the industry, and recognition of common requirements. The group agreed that security, interoperability, and integration would be essential for adoption. Going forward, a consortium of the meeting members will define the scope and implementation of a smart contracts protocol layer across several major blockchain systems.

Imagine a world where your washing machine can detect when you are running low on detergent, automatically engage with the market, negotiate the best price, and reorder the necessary product. The same goes with items in your refrigerator—milk, eggs, and so on. These smart devices will know your calendar and put off reordering if you are on vacation or away from home for a while. We're sure there will be some teething trouble with the first iterations of these devices and the smart contracts associated with them. As early adopters of this technology, we can imagine coming home to a pile of large boxes of fabric softener, enough to last a lifetime. But these devices and their associated intelligence will iterate and get it right for the mass market.

Summary

While blockchain technology is at the heart of cryptocurrencies like Bitcoin and Ethereum, and what they can do as decentralized, stateless currency and payment platforms, it clearly is a technology with widespread applicability in many sectors of business. This chapter gave some depth to real-world use cases in the financial services industry. These use cases ultimately lead to faster throughput, reduced costs, improved accuracy, greater transparency, and quality, reliability, simplicity, and traceability. The chapter also discussed smart property and smart contracts and showed how they can be used in conjunction in the not too distant

world of blockchain technology. The chapter closed with how blockchain and the Internet of Things (IoT) will have a consortium of startups and companies to help define and refine security, interoperability, management, and coordination between connected devices. While IoT is still in its early stages of evolution and is currently composed mostly of technologies that either collect data or allow remote monitoring and control, this will change as devices become smarter and artificial intelligence is added. This network of things will transition toward becoming a network of autonomous devices that talk to each other and—hopefully—make smart decisions without the need for human intervention or interpretation. In short, we live in exciting times for blockchain technology.

3 Technology Use Cases

In the 1970s, the Internet was a small, decentralized collective of DARPA computers, called ARPANET. The personal-computer revolution that followed built upon that foundation, stoking optimism encapsulated by John Perry Barlow's 1996 manifesto "A Declaration of the Independence of Cyberspace" (www.eff.org/cyberspace-independence). Barlow described a chaotic digital utopia, where "netizens" self-govern and the institutions of old hold no sway. "On behalf of the future, I ask you of the past to leave us alone," he writes. "You are not welcome among us. You have no sovereignty where we gather."

This is not the Internet we know today. Two decades later, the vast majority of communications flow through a set of central servers run by a small group of corporations under the influence of those companies and other institutions. Netflix, for instance, now comprises 40 percent of all North American Internet traffic. Engineers anticipated this convergence. In the late 1960s, key architects of the system for exchanging small packets of data that gave birth to the Internet predicted the rise of a centralized "computer utility" that would offer computing much the same way that power companies provide electricity. Today, that model is largely embodied by the cloud-computing giants Amazon, Google, Azure, and other cloud-computing companies. They offer convenience at the expense of privacy. Internet users now regularly submit to terms-of-service agreements that give companies a license to share their personal data with other institutions, from advertisers to governments. In the United States, the Electronic Communications Privacy Act of 1986 (ECPA), a law that predates the Web, allows law enforcement to obtain without a warrant private data that citizens entrust to third parties, including location data passively gathered from cell phones and the contents of emails that have either been opened or left unattended for 180 days. Note that under the ECPA only a subpoena or an 18 U.S.C. §2703(d) order with little judicial review is needed to allow access to the aforementioned private data. As Edward Snowden's leaks have shown, this massive information set allows intelligence agencies to focus on just a few key targets in order to monitor large portions of the world's population. The National Security Agency (NSA) wiretaps the connections between data centers owned by Google and Yahoo, allowing the agency to collect users' data as it flows across the companies' networks. A lack of trust surrounds the U.S. cloud industry. The NSA collects data through formal arrangements with tech companies, ingests web traffic as it enters and leaves the United States, and deliberately weakens cryptographic standards.