Blockchaintechnologie als Plattform für Smart Contracts

Präsentiert von Thees und Christian

Gliederung

1. Grundlagen
    a. Smart Contracts
    b. Blockchain
2. Smart Contracts mittels Blockchaintechnologie
    a. Exkurs: Ethereum
    b. Bedeutung von Smart Contracts auf Basis von Blockchaintechnologie
3. Grenzen und Lookahead
    a. Rechtliche Aspekte
    b. Sicherheitsaspekte
    c. Oracles
    d. Mögliche Anwendungsbereiche

Smart Contract

“Smart Contracts sind Computerprotokolle, die Verträge abbilden oder überprüfen oder die Verhandlung bzw. Abwicklung eines Vertrags technisch unterstützen.”

- Begriff 1994 durch Nick Szabo geprägt

Smart Contract

“Digitaler Verkaufsautomat”

User Input -> Output

Smart Contract

„Das Eigentum geht mit vollständiger Bezahlung über“

„if ($AmountReceived >= $Price) $OwnerDB[$AssetID] = $BuyerID“.

Smart Contract

Blockchain-Technologie

- Erwartungen: disruptive “next big thing”-Technologie vs. Überschätzung der Marktreife und praktischer Anwendung.
- **eine Blockchain ist eine Datenbank**, vorgehalten von mehreren Teilnehmern („Nodes“) eines Netzwerkes (bestehend aus Clients und sog. Minern).
- Integrität mittels einer fortlaufenden Prüfsumme sowie durch eine kryptographische Verrechnung gewährleistet.
- Gegenüber konventionellen Datenbanksystemen: keine nachträgliche Änderung aufgenommener Informationen.
    - Gemeinsame und sichere Nutzen von Daten soll ermöglicht werden..

Ziele der Technologie

- Durch **Transparenz und Dezentralisierung** werden Intermediäre von Transaktionen, wie etwa Banken, überflüssig.
- Potentielle Anwendungsbereich der Blockchain-Technologie geht aber weit über das Finanzwesen hinaus. Er erfasst insbesondere auch Versicherungen, Medien, den Energiesektor sowie die öffentliche Verwaltung, darunter z.B. staatliche Register wie das Grundbuch.
    - Zukunftsidee: Blockchain öffentliche Verzeichnisse, die sowohl fälschungssicher als auch unabhängig von zentralen, kontrollierenden Instanzen sind.

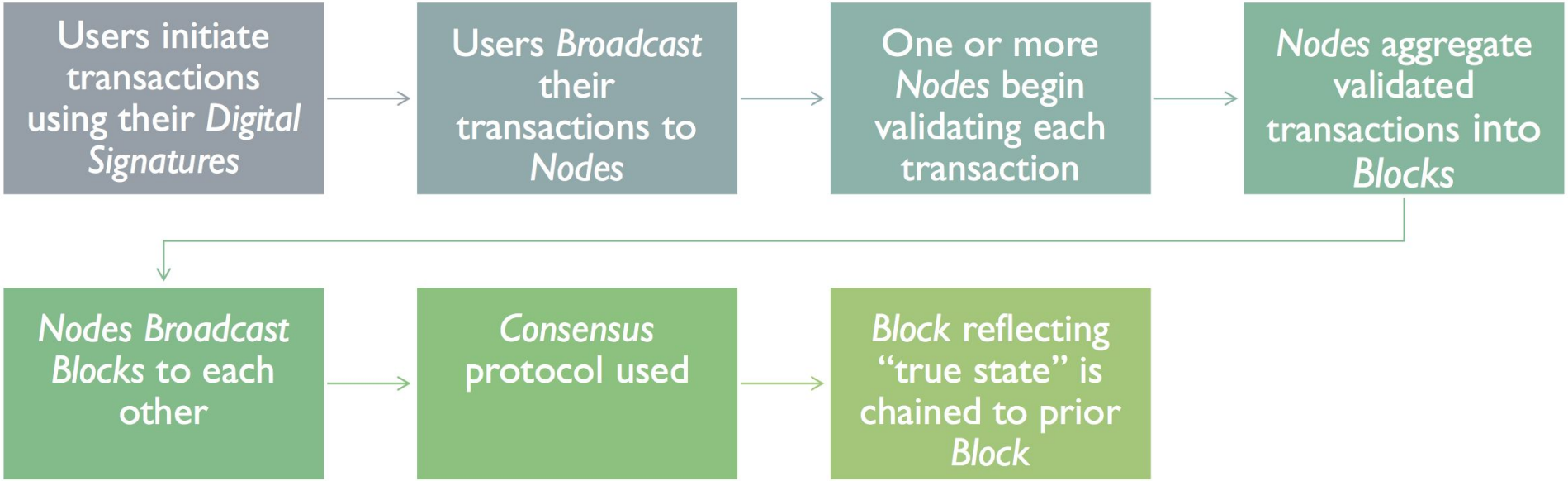
Distributed-Ledger-Technologie

- Blockchain ist eine Distributed Ledger Technology
- Übersetzt: verteilte Datenbanktechnologie.
- Die Blockchain kann man sich als eine Liste aller in einem Peer-To-Peer-Netzwerk (P2P-Netzwerk) vorgenommenen Transaktionen vorstellen.
- Jeder Teilnehmer, der die zur Nutzung nötige Software bei sich verwendet, nimmt am Netzwerk teil.
- Das Netzwerk setzt sich daher aus allen teilnehmenden Rechnern („Nodes“) zusammen, die über das Internet miteinander kommunizieren, ohne dass dabei eine zentrale Verwaltung nötig wäre.

Distributed-Ledger-Technologie

- Eine Speicherung der Blockkette mit allen Transaktionen zwischen den Nutzern des Netzwerkes erfolgt für gewöhnlich bei allen Nodes
- Eine Transaktion kann dabei jede Art von Information sein.
- Die Liste der Transaktionen wird in Blöcken fortgeschrieben, die aufeinander aufbauen und jeweils an den vorherigen Block angehängt werden.
- Jeder neue folgende Block ist mit den vorherigen mathematisch verbunden, sodass eine Kette von Blöcken, die Blockchain, entsteht.

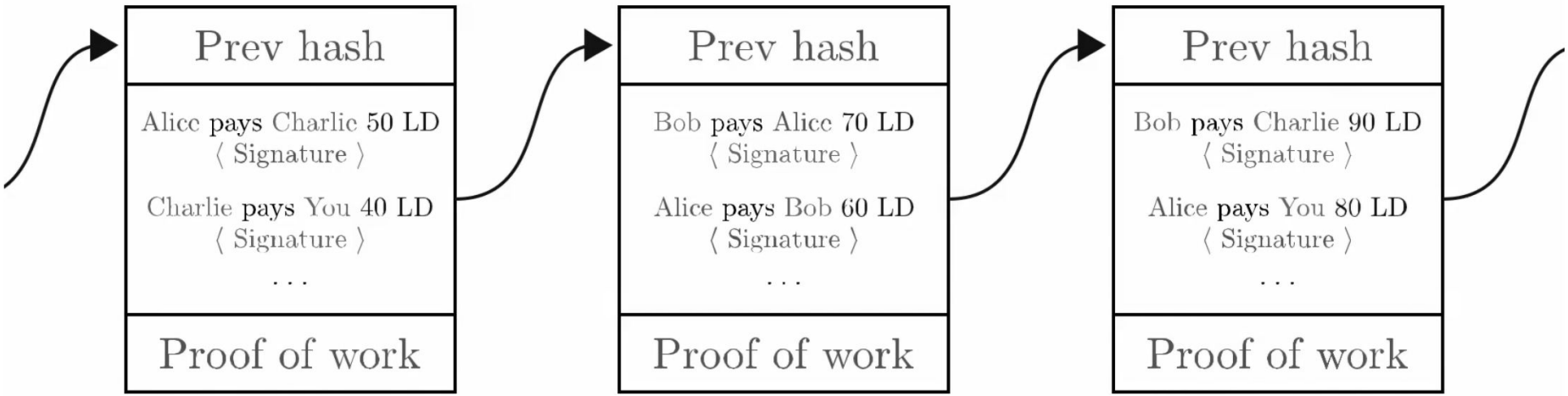
Distributed-Ledger-Technologie



Anwendung kryptographischer Verfahren

Initiation and Broadcasting of Transaction	• *Digital Signatures* • *Private/Public Keys*
Validation of Transaction	• *Proof of Work and certain alternatives*
Chaining Blocks	• *Hash Function*

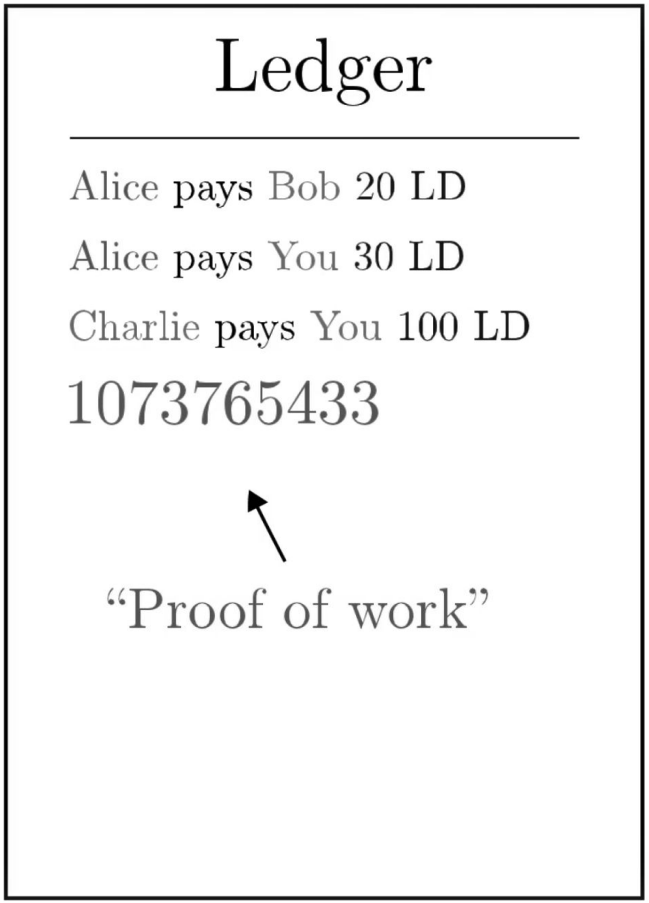
Blockchain


Prev hash
Alice pays Charlie 50 LD
⟨ Signature ⟩
Charlie pays You 40 LD
⟨ Signature ⟩
. . .
Proof of work
Prev hash
Bob pays Alice 70 LD
⟨ Signature ⟩
Alice pays Bob 60 LD
⟨ Signature ⟩
. . .
Proof of work
Prev hash
Bob pays Charlie 90 LD
⟨ Signature ⟩
Alice pays You 80 LD
⟨ Signature ⟩
. . .
Proof of work

Mining

- Dabei werden Transaktionen verifiziert und zu Blöcken aggregiert. Als Gegenleistung erhalten die Miner neu geschöpfte Bitcoin-Einheiten gutgeschrieben.

Proof of work



SHA256

Probability: $\frac{1}{2^{30}} \approx \frac{1}{1{,}000{,}000{,}000}$

30 zeros?

000000000000000000000000000011
001100010110111011001001001101 10
100000000100011001011011010001 1
101111111001110001100100101110 00
110110111011100101011011010001 11
000111100010000010001001100001 10
111001110001101000011000100100 01
100001011000100110100001010000 00

Proof of work

Ledger

Alice pays Bob 20 LD

Alice pays You 30 LD

Charlie pays You 100 LD

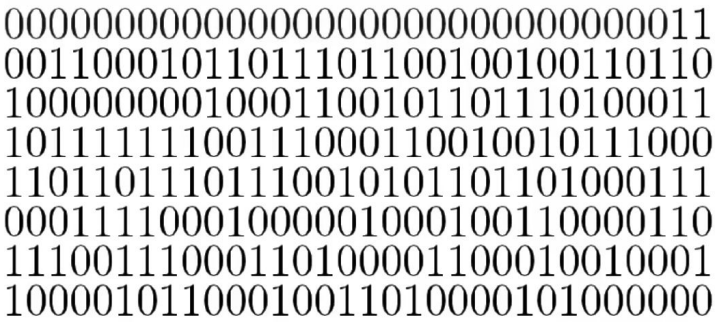
1073765433

"Proof of work"

SHA256

Probability: $\frac{1}{2^{30}} \approx \frac{1}{1{,}000{,}000{,}000}$

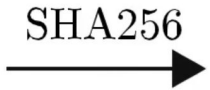
30 zeros?



Blockchain

M1

Prev hash
Miner 1 gets 10 LD Alice pays Bob 20 LD Charlie pays You 50 LD ⋮
2424597031397468




```
10001010110111110110100000011010
10111111000110010110000000100000
00101000110000011101001101010000
00110001001000011111100110011001
11000000110011110010110011100010
11001001111110011101101010101001
11000000110010000101000111010111
10111100000010100110011010110111
```

M2

Prev hash
Miner 2 gets 10 LD Alice pays Bob 20 LD Charlie pays You 50 LD ⋮
7408300263047858


```
10001010110111110110100000011010
10111111000110010110000000100000
00101000110000011101001101010000
00110001001000011111100110011001
11000000110011110010110011100010
11001001111110011101101010101001
11000000110010000101000111010111
10111100000010100110011010110111
```

M3

Prev hash
Miner 3 gets 10 LD Alice pays Bob 20 LD Charlie pays You 50 LD ⋮
4635005509269886


```
10001010110111110110100000011010
10111111000110010110000000100000
00101000110000011101001101010000
00110001001000011111100110011001
11000000110011110010110011100010
11001001111110011101101010101001
11000000110010000101000111010111
10111100000010100110011010110111
```

Dezentralisierung

- **Physisch strukturelle Dezentralisierung** — aus wie vielen **physischen Computern** besteht ein System? Wie viele Computerausfälle können toleriert werden?
- **Politische Dezentralisierung**—wie viele **Individuen oder Organisationen** haben Kontrolle über das System aus Computern?
- **Logische Dezentralisierung** — Sind Interface und Datenstrukturen ein einziges einheitliches Objekt, oder amorpher Schwarm? Als simple Heuristik: spaltet man das System in der Hälfte (Anbieter und Nutzer inbegriffen, können beide Hälften unabhängig weiter funktionieren?

Dezentralisierte Blockchain

- Blockchain als politisch dezentralisiert (keine zentrale Kontrolle)
- Physisch strukturell dezentralisiert (kein infrastruktureller Schwachpunkt
- **Logisch zentralisiert** (ein allgemein akzeptierter Zustand, das System verhält sich wie ein großer Computer)

Transparenz oder Datenschutz?

- Öffentlich einsehbares, dauerhaftes und unveränderliches Register
- erscheint gegensätzlich zu Grundsätzen der Datensparsamkeit, Datenvermeidung und Zweckbindung.
- Adressat und Absender sind allerdings pseudonym
- Chancen für Datenschutz ergeben sich durch kryptographische Mechanismen und “Privacy by default”
-

Smart Contract mittels Blockchain

- Smart Contract wird auf der Blockchain gespeichert und seine Inhalte von den Minern ausgeführt -> Resultat wird wiederum auf der Blockchain gespeichert
- Transaktionsgebühren bemessen sich in den meisten Systemen an der Komplexität des auszuführenden Codes
- Ermöglicht Peer-to-Peer-Transaktionen ohne einen zentralen Server im Mittelpunkt

Exkurs: Ethereum

- Relevanteste Smart Contract Plattform
- Nutzungsbereiche (neben regulären Transaktionen) vor allem innerhalb des E-Votings, Crowdfundings, etc.
- Abwicklungen mittels interner Kryptowährung Ether (ETH)
- Smart Contracts werden hauptsächlich in der eigens entwickelten Programmiersprache Solidity verfasst

Exkurs: Ethereum - Solidity

```
contract TestCoin {
    // Das Schlüsselwort "public" erlaubt es auch anderen Smart Contracts, die Variable auszulesen.
    // Diese Variable hat den Typ "address" und steht für die Adresse eines Ethereum Accounts.
    address public minter;
    mapping (address => uint) public balances;

    // Events erlauben es Ethereum-Clients, auf Ereignisse des Contracts zu reagieren.
    event Sent(address from, address to, uint amount);

    // Die Funktion mit demselben Namen wie der Smart Contract ist der Konstruktor.
    // Diese Funktion wird ein einziges Mal bei der Erstellung des Contracts aufgerufen.
    function TestCoin() {
        minter = msg.sender;
    }

    // Mit dieser Funktion kann der "minter" Ethereum Account anderen Accounts einen beliebigen
    // Betrag des "TestCoin" überweisen.
    function mint(address receiver, uint amount) {
        if (msg.sender != minter) return;
        balances[receiver] += amount;
    }

    // Diese Funktion erlaubt es Ethereum-Accounts, sich gegenseitig TestCoin zu überweisen.
    function send(address receiver, uint amount) {
        if (balances[msg.sender] < amount) return;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        Sent(msg.sender, receiver, amount);
    }
}
```

Smart Contracts on the Blockchain

- **“Immutability" –** Blockchain als unveränderbare Plattform garantiert Unverfälschtheit
- **“Disintermediation” –** Ermöglicht es Parteien, Vereinbarungen mit einer geringeren Abhängigkeit von Zwischenhändlern abzuschließen
- **“Transparenz” –** Erzeugt eine Umgebung mit Vertrauen, da die Logik und Informationen im Vertrag für alle Teilnehmer im Blockchain-Netzwerk sichtbar sind
- **Vertrauen –** Man muss seinem Vertragspartner nicht vertrauen, weil dieser die Durchführung des Vertrages gar nicht unterlassen kann, sobald die Zahlung eingegangen ist.

Rechtliche Aspekte

- Unbeschränkter Einsatzbereich schließt ein konkretes Blockchain-Gesetz aus - Analyse der Anwendung und Inanspruchnahme jeweiliger Technik.
- Problematik der Haftung: Fehler im Code können schnell teuer werden.
  - Freizeit Programmier: Haftung im Falle grober Fahrlässigkeit oder Unterschlagung bekannter Programmierfehler
  - Gewinnorientierte Programmierer: Produktionshaftungsgesetz
- Noch kein Präjudiz für aktuelle Rechtslage, Validität schwierig

Rechtliche Aspekte

- Ethereum wälzt per AGB sämtliche Risiken auf Nutzer : https://www.ethereum.org/agreement
- Anbieter Versteckspiel: rechtliche Verfolgung durch Pseudonymität schwierig
- Quantität der möglichen Vertragspartner
- Juristische Verträge auf der Blockchain speichern: nur bei öffentlichen Registern ist Publizität gewünscht

Sicherheitsaspekte

- **Genauigkeit –** Da es sich bei einem Smart Contract um ein Computerprogramm handelt, muss jede Vertragsbedingung exakt codiert sein. Es besteht die Möglichkeit von Fehlinterpretationen bzw Versäumnissen durch den Programmierer, was zu Vertragslücken führen kann.
- **Unsachgemäße Eingaben/Ungültige Vertragsgrundlagen –** kann zu falschen Verträgen oder zur Nichtausführung von Verträgen führen. Im Falle eines traditionellen Vertrags können die Parteien den Rechtsweg gehen. Bei Smart Contracts ist dies aufgrund der anhaltenden Debatte über ihre Rechtsfähigkeit noch nicht möglich.
- **Bugs und Fehler im Code/Sicherheitslücken –** Es besteht die Möglichkeit von Verfahrensschwierigkeiten auf Grundlage der Frage nach Verantwortlichkeit und Haftung für diese Fehler. Sie könnten außerdem zu unvorhergesehenen Problemen führen, wie beispielsweise Hackerangriffe, die auf diese Fehler abzielen. So konnten Hacker im Juni 2016 bei einem Angriff auf die Decentralized Autonomous Organisation (DAO) 50 Millionen Ether erbeuten.

Oracles

- Unabhängige “Quelle der Wahrheit” überprüft und verifiziert den Status von Bedingungen, die für die Vertragsablauf relevant sind.
- Bei einem Vertrag, der ausschließlich auf digitaler Information basiert, ist dies einigermaßen möglich.
- Wenn Bedingungen aus der realen Welt erfüllt werden müssen, benötigt man also eine möglichst objektive Informationen.

Oracles

- **Software Oracles**
  - Verarbeiten Informationen, die online zu finden sind.
  - **Beispiel:** Temperaturen, Preise von Handelswaren, Flügen oder Zugverspätungen. Als Quelle dienen hier beispielsweise Firmen.
- **Hardware Oracles**
  - Liefern Informationen, die aus der physischen Welt stammen
  - **Beispiel:** Ein Fahrzeug, dass durch eine Sensorschranke fährt und die Daten an den Smart Contract weiterleitet.
- **Consensus Based Oracles**
  - Es werden mehrere Oracles kombiniert, um sich nicht auf eine einzige externe Quelle zu verlassen. Es wird Konsens gebildet, um Entscheidungen zu treffen.
  - **Beispiel:** Prognosemärkte

Oracles

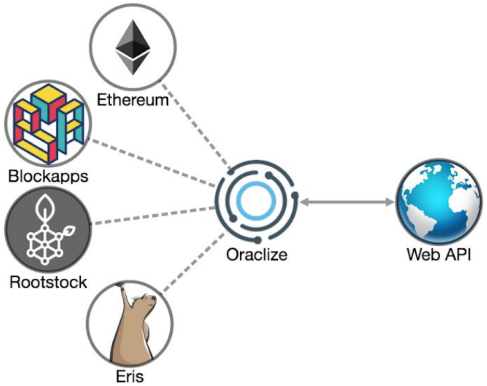
HOME HOW IT WORKS PROJECTS ABOUT CONTACT US MORE +

HOW IT WORKS

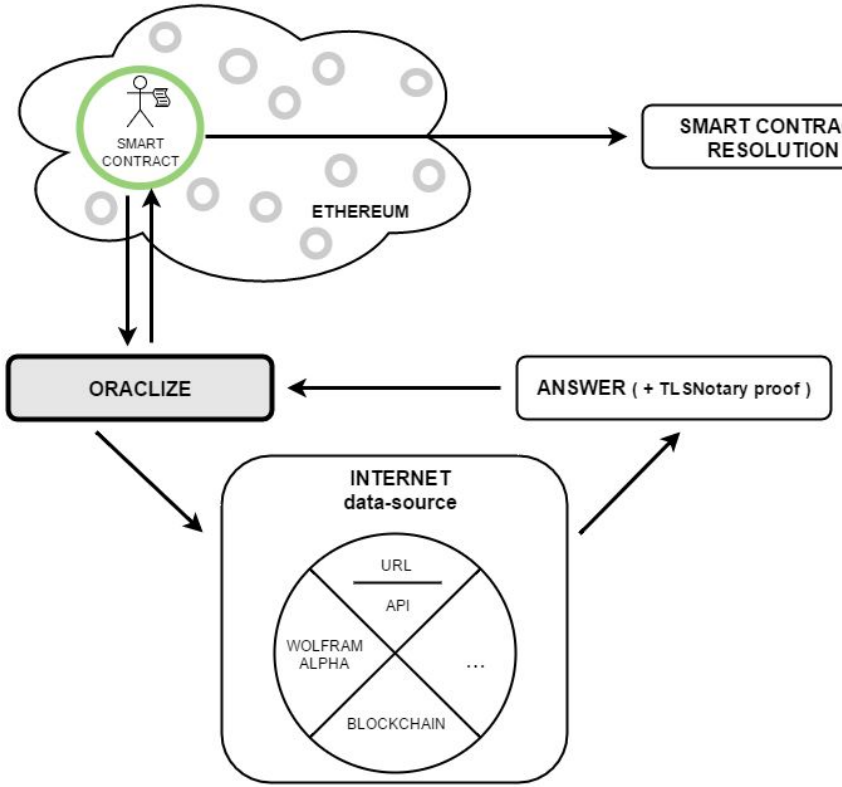
data carrier for decentralized apps

Smart contracts live like in a walled garden,
they cannot fetch external data on their own.
Oraclize is here to help. We act as a data carrier,
a reliable connection between Web APIs and your Dapp.
There is no need to open additional trustlines as
our good behaviour is enforced by cryptographic proofs.

DISCOVER MORE



Oracles



1. **“Why would I trust a data-source?** Most of the times you shouldn’t. Finding a consensus on different data-sources is a good way to go and gives you extra reliability while still using somehow “centralized” (useful) data.”
2. **“What if the oracle/oracle network gives me back a wrong result?** This is the main point around preferring to use an oracle network consensus instead of a single oracle.”

Lookahead - mögliche Anwendungen am Beispiel Smart Contracts und IPRs

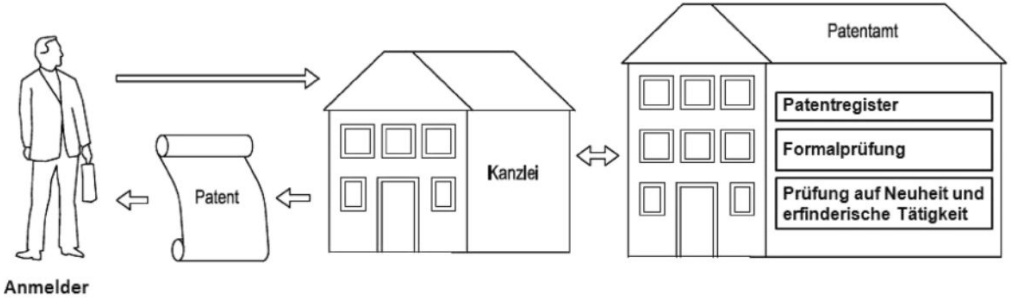


Abb. 1 Verwaltung von IPRs mit einer Kanzlei

Lookahead - mögliche Anwendungen am Beispiel Smart Contracts und IPRs

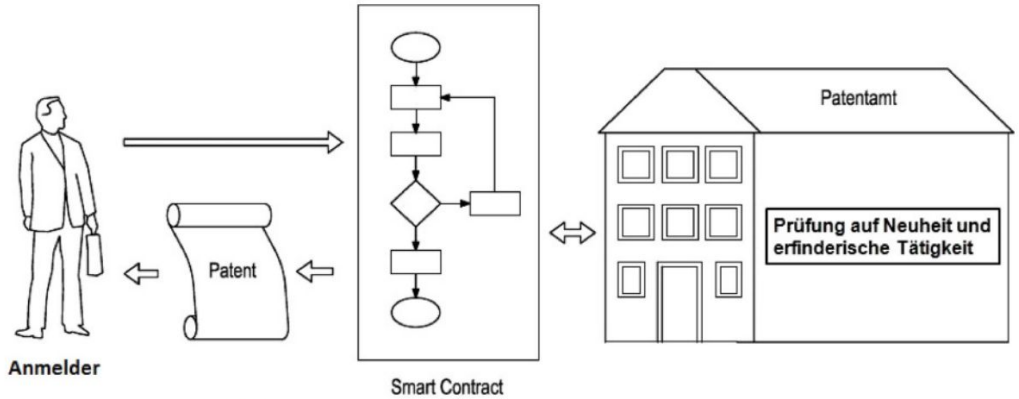


Abb. 2 Verwaltung von IPRs mit Smart Contracts

Diskussion