Sonderdruck aus JavaSPEKTRUM 03/2017

Kontrolle ist gut, Vertrauen ist besser

So könnte der öffentliche Sektor von Blockchain profitieren

Jakob Boos, Carmen Eisenacher, Andreas Lutz, Lars Santesson, Christof Tinnes

Blockchain erobert nicht nur die Finanzwelt, auch im öffentlichen Sektor steigt das Interesse an dieser Technologie. Die Stärke der Blockchain zeigt sich hier insbesondere, wenn die Kontrolle über Daten abgegeben oder geteilt werden soll. Die Anwendungsgebiete sind dabei vielseitig, wie der Artikel in zwei Szenarien herausarbeitet. Außerdem zeigen praktische Kurzanleitungen am Beispiel des IBM Bluemix Blockchain Service und von Tendermint, wie Entwickler entsprechende Java-Anwendungen mit beherrschbarem Aufwand realisieren können.

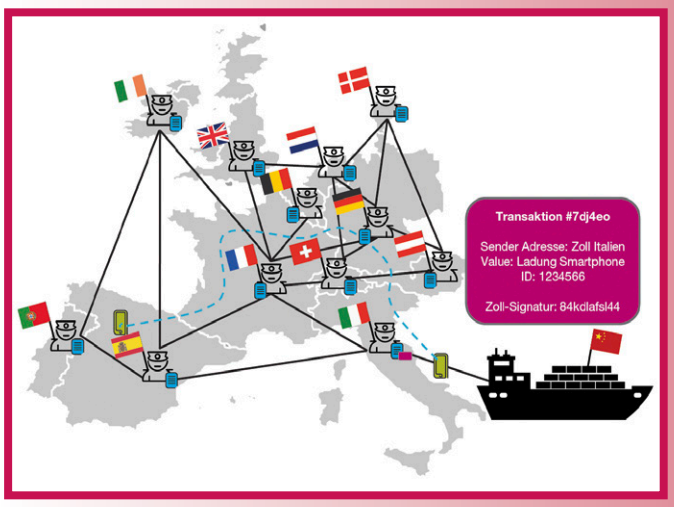

Abb. 1: Rückverfolgung einer Smartphone-Lieferung

Laut Gartner befindet sich die Blockchain-Technologie am Höhepunkt des Hype Cycles (vgl. [Gart]). Wie aber sieht es damit im öffentlichen Sektor aus? Im Kern ist die Blockchain-Technologie dezentral orientiert. Gerade deshalb scheint ihr Einsatz für die föderale und nach dem Ressortprinzip organisierte öffentliche Verwaltung in Deutschland vorteilhaft. Mit Hilfe von ausschließlich dezentralen Instanzen können Informationen zwischenbehördlich oder zwischen Bürgern, Unternehmen und Behörden ausgetauscht werden. Gleichzeitig bietet die Technologie eine hohe Sicherheit vor Korruption oder Angriffen von außen sowie eine hohe Verfügbarkeit.

Anhand von zwei Szenarien arbeiten wir im Folgenden eine mögliche Nutzung von Blockchain und den damit verbundenen Chancen genauer heraus.

Szenario 1: Validierung von Dokumenten

Jeder in Deutschland hat sich vermutlich schon einmal über Behördengänge geärgert, bei denen es darum geht, beglaubigte Kopien von der Behörde A zur Behörde B zu bringen. Dennoch bieten nicht alle Standesämter einen (Online-)Service an, um etwa eine Geburtsurkundenabschrift postalisch zu beantragen.

Behörden können zwar Dokumente mit elektronischen Signaturen versehen; jedoch müssen zum Beispiel bei einer Qualifizierten Elektronischen Signatur dedizierte Signaturkarten, Lesegeräte und Software vorgehalten werden, und nicht alle Dokumentenformate können unterstützt werden. Eine weitere Möglichkeit: Verwaltungsmitarbeiter hinterlegen kryptografische Prüfsummen wie SHA-256 der Dokumente in einer zentralen Datenbank, sodass andere Behörden digitale Dokumente dann über eine Hashberechnung und einen Datenbank-Lookup überprüfen können. Doch hier stellt sich schnell die Frage, wer die Datenhoheit über die Datenbank hat. Länder oder Kommunen haben in der Regel erhebliche Vorbehalte, die Datenhoheit an eine Bundesbehörde abzugeben.

Genau hier kann die Blockchain-Technologie helfen. Sie löst das Problem einfach dadurch, dass sich die Behörden die Datenbank „teilen“. Behörden laden in der Blockchain den Dokumentenhash, Dokumententyp und den Namen des Bürgers/Unternehmens hoch. Damit können Behörden und andere Beteiligte einfach digitale Dokumente auf Echtheit überprüfen. So kann ein Standesamt den Dokumentenhash einer digitalen Geburtsurkunde hochladen und andere Standesämter können diese digitale Geburtsurkunde validieren, indem sie nach dem Hashwert des zu prüfenden Dokuments in der Blockchain suchen. Durch das dezentrale Netzwerk werden Behördengänge so in puncto Dokumentenvalidierung deutlich einfacher, sicherer und moderner.

Die Lösung ist nicht nur auf Dokumente beschränkt: Letztendlich bietet die Blockchain eine ideale Möglichkeit, Signaturen öffentlich oder in einem beschränkten Teilnehmerkreis zu teilen. Für SSL oder andere auf PKI-basierende Verfahren könnte die Blockchain-Technologie daher Vertrauen in zentrale Zertifizierungsstellen überflüssig machen.

Szenario 2: Logistik und Zoll

Im Bereich Logistik könnte die Blockchain-Technologie zur Überwachung von Lieferketten eingesetzt werden. Die Vorteile: Illegale Einfuhren von Waren und Markenpiraterie fallen schneller auf und Verbraucher können exakt nachvollziehen, woher ihre Waren kommen. Auch können Regulatoren überprüfen, ob gefährliche Produkte wie Chemikalien oder Waffen über sichere und korrekte Wege transportiert wurden.

Das Beispiel der Europäischen Union (EU) illustriert, wie die Blockchain-Technologien Behörden über föderale Ebenen und über Ländergrenzen hinweg in einem dezentralen System miteinander verknüpfen können. Ein typisches Problem bei dezentralen Vereinigungen ist, dass kein Teilnehmer Datenhoheiten an einen anderen abtreten möchte. Dies gilt auch für die EU-Mitglieder. Insbesondere kleinere Länder fürchten, dass über Zentralisierungen eine irreversible Abhängigkeit entsteht. Dezentrale Verzeichnisse schaffen hier Abhilfe über ein gemeinsames, grenzübergreifendes System, in dem alle Beteiligten gleichberechtigte Partner sind. Die Blockchain-Technologie kann insofern die Basis für einen schnelleren und besseren Datenaustausch sein.

In einem fiktiven Anwendungsfall sind Zoll-Institutionen Teilnehmer in einem Blockchain-Netzwerk. Entscheidungen, wie die Aufnahme neuer Teilnehmer, stimmen alle Beteiligten in einem dezentralen Verfahren ab (Konsensfindung). In diesem Netzwerk können je nach Bedarf alle betroffenen Institutionen innerhalb und gegebenenfalls außerhalb der EU herange-

zogen werden. Eine Kopie des Teilnehmerverzeichnisses liegt allen Teilnehmern des Netzwerks vor. Kommen weitere Akteure zum Netzwerk hinzu, bieten sich Konsens-Mechanismus-Protokolle an, die auf flexiblem Vertrauen basieren. Das heißt: Neue Mitglieder werden über bekannte, bereits angemeldete Mitglieder eingegliedert. Eine zentrale Instanz für das Hinzufügen neuer Teilnehmer entfällt damit.

Einen beispielhaften Prozess zeigt Abbildung 1 in stark vereinfachter Weise: Ein Frachtschiff mit einer Ladung Smartphones aus China erreicht die europäische Grenze an einem italienischen Hafen. Die italienische Zollstelle registriert die Ladung und schreibt anschließend eine Transaktion in eine Zoll-Blockchain. Inhalte der Transaktion sind unter anderem die Adresse der Zollstelle „Zoll Italien", der Wert „Ladung von Smartphones", eine eindeutige Identifikationsnummer sowie eine digitale Signatur der Zollstelle.

Nach der Registrierung auf der Blockchain wird die Ladung per LKW über Österreich, Deutschland und Frankreich zu ihrem Zielort in Spanien transportiert. An Kontrollstellen können Mitarbeiter schnell feststellen, über welchen Grenzübergang und in welcher Reihenfolge die Ware nach Europa kam und ob bereits Zollgebühren erhoben wurden. Eine illegale Einfuhr wird so schnell erkennbar und die Verantwortlichen können Strafmaßnahmen ergreifen. Auch kann der Empfänger/Käufer am spanischen Zielort überprüfen, ob und wann die Ware ordnungsgemäß nach Europa eingeführt wurde und wie lange sie unterwegs war. Gerade bei verderblichen Gütern ist dies wichtig.

Entscheidungshilfe: Blockchain – ja oder nein?

So vielversprechend Blockchain in vielen Fällen ist, sie ist nicht die Lösung für alle Probleme. Ihr Einsatz will gut durchdacht sein – auch, um voreilige Schlüsse zu vermeiden. Die folgende Entscheidungshilfe unterstützt bei den ersten Überlegungen zum Blockchain-Einsatz:

Sind Blockchain-Technologien fachlich geeignet? (Knock-out-Kriterien)

Alle hier aufgeführten Kriterien müssen erfüllt werden, um Blockchain näher in Erwägung zu ziehen. Ist ein Kriterium nicht erfüllt, sind aller Voraussicht nach alternative Lösungen einfacher und wirtschaftlicher:

- *Muss ich für die Wahrnehmung einer Aufgabe gemeinsam mit anderen Beteiligten außerhalb meiner Organisation (und Kontrolle) Informationen verfügbar machen?* Beim Beispiel der Dokumentenvalidierung sind wegen des behördenübergreifenden Austausches behördenspezifische Dokumentenvalidierungslösungen keine Option. Die Speicherung in einer zentralen Datenbank über alle föderalen Grenzen hinweg ist ebenfalls keine Alternative – wegen der Unabhängigkeit und der fehlenden Akzeptanz der jeweiligen Behörden. Die Speicherung der Prüfsummen in einer verteilten Blockchain durch gleichwertige Beteiligte hingegen schon.
- *Müssen mutwillige oder unfreiwillige Manipulationen der Informationen verhindert werden und muss nachgewiesen werden, dass keine Manipulationen möglich waren?* Das Beispiel der Dokumentenvalidierung zeigt: Es können sehr viele Behörden mit unterschiedlichen Interessen involviert sein. Behörden, die Dokumente erstellen, wollen verhindern, dass andere Behörden oder Bürger ihre erstellten Dokumente eventuell manipulieren. Auch bei digitalen Wahlen kann Vertrauen geschaffen werden, indem eine unabhängige und nicht manipulierbare Infrastruktur genutzt wird.
- *Ist eine dezentrale Administration möglich?* In manchen Fällen existieren eventuell Beschränkungen (zum Beispiel gesetzlicher Art), die eine teilweise Abgabe von Kontrolle über Informationen verhindern.

Sind Blockchain-Technologien fachlich optimal? (Begünstigende Kriterien)

Wird folgendes Kriterium erfüllt, ist der Einsatz von Blockchain besonders vorteilhaft. Gegebenenfalls werden sich im Laufe der Zeit weitere Kriterien herauskristallisieren:

- *Muss ein bestimmter Ablauf/Reihenfolge sichergestellt oder im Nachgang nachgewiesen werden?* Bei der oben beschriebenen Prüfung des Transports von Smartphones in die EU kann die Blockchain die Transportreihenfolge nachweisen. Die entsprechende Zoll-Anwendung kann während des Transports mit der Hilfe von Blockchain die Einhaltung von Regeln hinsichtlich erlaubter Transportstrecken sicherstellen. Zudem bieten viele Blockchain-Technologien die Möglichkeit, solche Überprüfungen oder ganze Transportstrecken schon in die Blockchain zu codieren.

Gibt es für den Einsatz unüberwindbare technische Restriktionen? (Ressourcenabhängige Kriterien)

- *Sind die Datenvolumen beziehungsweise das Datenwachstum für alle Teilnehmer verträglich?* Bei großen Datenmengen sind Blockchain-Technologien unter Umständen nicht geeignet, da die Daten redundant im Netzwerk verteilt und gespeichert werden müssen. Bei der Dokumentenvalidierung sollten die jeweiligen Verantwortlichen daher etwa nicht alle Informationen in der Blockchain speichern, sondern nur sichere Prüfsummen.
- *Sind die Latenzzeiten bei Informationsspeicherungen mit allen Interessen der Teilnehmer verträglich?* Der Preis für die Aushandlung von Konsens in der Blockchain ist unter anderem, dass man höhere Latenzen in Kauf nehmen muss. Passt das nicht zur fachlichen Anforderung, gilt es eine andere Technologie zu suchen. Die verschiedenen Konsensmechanismen unterscheiden sich – mit Blick auf die Latenzen – in der Regel auch stark.

Diese beiden technischen Beschränkungen sind bei jedem Einsatz von Blockchain-Technologien zu überprüfen. Sie hängen stark mit der Umsetzung der Anwendungsfälle und den gegebenen technischen sowie finanziellen Ressourcen zusammen.

How to

Blockchain-Anwendungen werden in einer Client-Server-Architektur realisiert (s. Abb. 2).

Will ein Architekt die Blockchain-Technologie in der Softwareentwicklung konkret einsetzen, sollte er eventuelle Sicherheitsrisiken bedenken. Danach kann man zum Beispiel Persistierung und Fachanwendung voneinander entkoppeln. Im Großen und Ganzen gibt es zwei Möglichkeiten: Entweder man sieht die Blockchain nur als einen Datenspeicher oder man integriert auch bestimmte Logiken, sogenannte Smart Contracts, in die Blockchain selbst. Anhand von zwei Beispielen werden wir dies skizzieren.

So funktioniert's mit dem IBM Bluemix Blockchain Service

Um einfach in die Blockchain-Programmierung einzusteigen, können Entwickler zum Beispiel mit dem IBM Bluemix Blockchain Service in wenigen Schritten ein Blockchain-Netzwerk aufsetzen und damit experimentieren. Der IBM Bluemix Blockchain Service ermöglicht eine einfache Entwicklung von interessanten Anwendungsfällen in einem sogenannten „per-

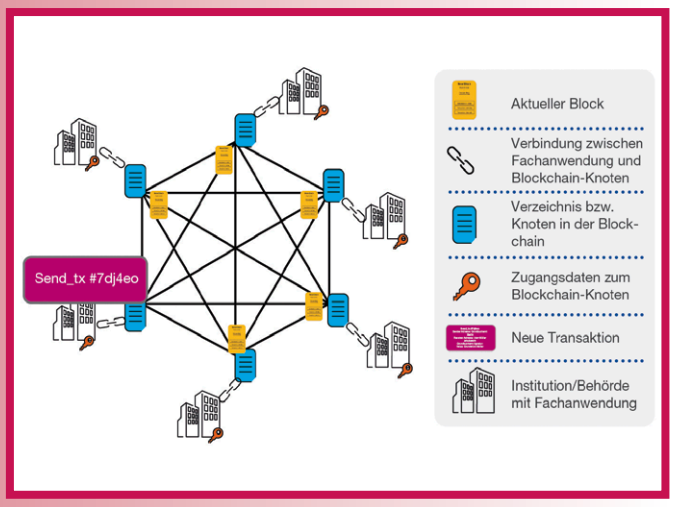

Abb. 2: Schematische Client-Server-Architektur einer Blockchain-Anwendung

missioned“ Blockchain-Netzwerk. Diese Art von Blockchains sind für eingeschränkte Teilnehmerkreise geeignet, wie etwa ein Netzwerk von öffentlichen Institutionen oder High-Security-Unternehmensnetzwerke. Der Service von IBM Bluemix basiert auf dem Open-Source-Projekt des Hyperledger der Linux Foundation (vgl. [IBM]).

Mit Hilfe des Service haben wir das beschriebene Szenario 1 „Validierung von Dokumenten“ prototypisch umgesetzt. Bei diesem Proof of Concept gibt es zwei Nutzergruppen, die Issuer (Behörden oder öffentlich-rechtliche Institutionen) und die User (Bürger oder Unternehmen). Eine Behörde (Issuer) kann einen Hashwert von einem beantragten Dokument mit Hilfe einer Programmierschnittstelle auf der privaten Behörden-Blockchain speichern. Dazu prüft die Applikation die Authentifizierung des Issuers. Innerhalb der Transaktion – in unserem Fall ist eine Transaktion die Speicherung des Dokumentenhashs – werden folgende Inhalte aufgenommen: der aktuelle Zeitwert, der Name des Issuers, der Name des Empfängers des Dokumentes und der

```
public DocumentEntity createDocument(
    byte[] fileContent, String issuer, String user, String fileName) {
  // Überprüfen, ob User eingeloggt ist
  if (user == null ||
      BlockchainApi.isRegistered(BlockchainApi.finde(user)) == false){
    return null;
  }
  // Überprüfen, ob Issuer eingeloggt ist
  if (issuer == null ||
      BlockchainApi.isRegistered(BlockchainApi.finde(issuer)) == false){
    return null;
  }
  Timestamp currentTimestamp =
    new Timestamp(Calendar.getInstance().getTime().getTime());
  DocumentEntity document =
    generateDocument(issuer, user, fileName,
                     currentTimstamp, fileContent)
  // gibt einen sicheren Hash z.B. SHA-256 zurück:
  String hash_Str = createHash(fileContent);
  document.setHash(hash_Str);
  // Dokumenteninformationen in separater Datenbank hinterlegen:
  save(document);
  // Rest Aufruf an die Blockchain:
  String[] args = {hash_Str, issuer, "0", user, };
  // neues Dokumentes mit Hash-Daten in der Blockchain:
  BlockchainApi.invokeFunc("init_doc",args);
  return document;}
```

Listing 1: Erzeugung und Persistierung eines Dokumentenhashs

Hashwert des Dokumentes. Über eine einfache Programmierschnittstelle werden die Inhalte nun auf der IBM Bluemix Blockchain gespeichert (s. Listing 1).

Nachdem der Hashwert des Dokuments hochgeladen wurde, können Privatpersonen oder Unternehmen über eine Weboberfläche das Dokument auf Echtheit überprüfen (s. Listing 2). Falls der Hashwert auf der Behörden-Blockchain gefunden wird, wird das Dokument als rechtsgültig anerkannt.

```
private String DocUser;

public Boolean validateDocument(byte[] fileContent,
    String user, String fileName) {
  // Überprüfen, ob User eingeloggt ist
  if(!BlockchainApi.isRegistered(BlockchainApi.find(user))){
    return false;
  }
  Timestamp currentTimestamp =
    new Timestamp(Calendar.getInstance().getTime().getTime());
  String[] hashArgument = createHashFast(fileContent);
  // REST-Request über API absenden:
  String response = BlockchainApi.queryFunc("read", hashArgument);
  return isHashValid(response);
}
```

Listing 2: Validierung eines Dokumentenhashs

In beiden Vorgängen – beim Hochladen des Hashwertes und beim Verifizieren des Dokumentes – wird ein REST-Aufruf an die IBM Bluemix Blockchain abgesendet. Die Kommunikation zwischen der Programmierschnittstelle, dem sogenannten Chaincode und der IBM Bluemix Blockchain wird in [GitH1] erklärt.

```
public String queryFunc( String funcToBeQueried, String[] args){
  if (this.chain.details.getDeployed_name() == null ){main();}
  boolean exists = queryExists(funcToBeQueried);
  if (!exists) {
    return null;
  }
  Options options = new Options(initRest);
  options.path = "/chaincode";
  Body body = setupRequestBody(funcToBeQueried, args);
  return restService.post(options, body);
}
```

Listing 3: Beispiel API-Call „queryFunc“

Mit Hilfe des IBM Bluemix Blockchain Service können wir also in wenigen Schritten einen Blockchain-Anwendungsfall aufsetzen und testen. Die Programmierschnittstelle ist eine einfache Möglichkeit, um Szenarien zu entwickeln und Geschäftslogiken in Form von Smart Contracts via Chaincode zu implementieren.

So funktioniert‘s mit Tendermint

Als zweites Beispiel verwenden wir Tendermints Application Blockchain Interfaces (ABCI) (vgl. [GitH2]) und dessen Java-Implementierung jABCI (vgl. [GitH3]), um zu zeigen, wie leicht die Blockchain-Technologie in der Praxis zu nutzen ist. Die Idee von Tendermint und des ABCI ist es, einen Knoten zu starten, der mit den anderen Netzwerkteilnehmern verbunden ist und sich hauptsächlich um Konsensbildung unter den Teilnehmern kümmert. Über das ABCI kommuniziert der Knoten dann mit einer Applikation. Dabei stellt die Applikation den Server des Interfaces dar und der Knoten den Client.

Im Wesentlichen muss die Applikation definieren, wie sie auf folgende Anfragen des Knotens antwortet:

- *CheckTx:* Transaktionen, die vom Knoten empfangen wurden, können durch die Applikation validiert werden. Der Knoten schickt die Transaktion bei negativem Ergebnis nicht weiter und würde sie nicht in seine Blöcke einbauen. Denkt man an einen konkreten Einsatz bei Wahlen, dann würde hier beispielsweise überprüft, ob ein „Wahl-Token" des Wahlberechtigten gültig ist und noch nicht verwendet wurde.
- *DeliverTx:* Transaktionen werden von der Applikation behandelt. Bei unserem fiktiven Wahlenbeispiel würden hier beispielsweise die Stimmen pro Partei zählen.
- Ein *„ProposeTx"* (dient in unserem Beispiel zur Abgabe der Wahlstimmen) bietet die Schnittstelle aktuell nicht. Dies geschieht über einen HTTP RPC am Blockchain-Knoten: `curl http://blockchainknoten:46657/broadcast_tx_sync?tx=\"transaktionsinhalt\"`

Der jABCI-Code sieht in etwa wie in Listing 4 aus, den Rest macht quasi der Tendermint-Core und jABCI.

```
public class ElectionHandler
    implements IDeliverTx, ICheckTx, ICommit {
  public static final String INVALID_VOTE_TEXT=
    "Partei nicht wählbar oder Stimme ungültig."
  [...]
  public ElectionHandler() throws InterruptedException {
    socket = new TSocket();
    socket.registerListener(this);
    new Thread(socket::start).start();
    while (true) {
      Thread.sleep(1000L);
    }
  }
  @Override
  public ResponseDeliverTx receivedDeliverTx(RequestDeliverTx req) {
    System.out.println("Eine Stimme wurde empfangen.");
    Vote vote = parseTransaction(req);
    if (isContainedInTokenList(vote.getToken()) &&
        isEligible(vote.getVote())) {
      return ResponseDeliverTx.newBuilder().
        setCode(CodeType.OK).build();
    }
    return ResponseDeliverTx.newBuilder().setCode(CodeType.BadNonce)
      .setLog(INVALID_VOTE_TEXT).build();
  }
  @Override
  public ResponseCheckTx requestCheckTx(RequestCheckTx req) {
    System.out.println("Eine Stimme wird validiert.");
    Vote vote = parseTransaction(req);
    if (!isContainedInTokenList(vote.getToken()) ||
        !isEligible(vote.getVote())) {
      return ResponseCheckTx.newBuilder().setCode(CodeType.BadNonce)
        .setLog(INVALID_VOTE_TEXT).build();
    }
    return ResponseCheckTx.newBuilder().setCode(CodeType.OK).build();
  }
  [...]
}
```

Listing 4: Beispiel-Implementierung „Election" des jABCI-Interfaces

Fazit

Die analysierten Szenarien zeigen deutlich die großen Chancen, die die Blockchain-Technologie im öffentlichen Sektor bietet. Ob zwischen Bürgern und Behörden, Behörden und Behörden oder Bürgern und Bürgern – im öffentlichen Sektor gibt es zahlreiche Abläufe, bei denen die Blockchain-Technologie existierende Hürden aus dem Weg räumen kann.

Mittels IBM Bluemix Blockchain Service und Tendermint können Entwickler entsprechende Lösungen schnell realisieren. Wir sind fest davon überzeugt, dass in nicht allzu ferner Zukunft die Blockchain zum Alltag unserer digitalen Behördengänge gehören wird.

Links

[Gart] Gartner's 2016 Hype Cycle for Emerging Technologies, 16.8.2016, `http://www.gartner.com/newsroom/id/3412017`

[GitH1] A tutorial to get you started with writing smart contracts for Hyperledger, `https://github.com/IBM-Blockchain/learn-chaincode`

[GitH2] Application BlockChain Interface, `https://github.com/tendermint/abci`

[HitH3] A Java implementation of the Tendermint Application BlockChain Interface, `https://github.com/jTendermint/jabci`

[IBM] `https://www.ibm.com/blockchain/offerings.html`





Jakob Boos ist Technical Account Manager bei Capgemini in Deutschland für den öffentlichen Sektor. Seine Themenschwerpunkte sind IT-Modernisierung sowie IT-Architektur und plattformbasierte Softwareentwicklung. E-Mail: jakob.boos@capgemini.com

Carmen Eisenacher schreibt aktuell ihre Masterarbeit zum Thema „Blockchain im öffentlichen Sektor" bei Capgemini. Sie beschäftigt sich mit neuen Technologien und innovativen Anwendungsfällen. E-Mail: carmen.eisenacher@capgemini.com

Andreas Lutz ist IT-Berater bei Capgemini in Deutschland. Seine Themenschwerpunkte sind Unternehmensarchitektur sowie Software-Engineering und moderne Webanwendungen.
E-Mail: andreas.lutz@capgemini.com

Lars Santesson ist Chefarchitekt bei Capgemini für Kunden im öffentlichen Sektor. Bei der Beratung seiner Kunden befasst er sich unter anderem mit der Frage, wie technologische Innovationen gewinnbringend bei Behörden eingesetzt werden können.
E-Mail: lars.santesson@capgemini.com

Christof Tinnes arbeitet als Software-Engineer bei Capgemini in Deutschland und forscht seit einem Jahr im Bereich Blockchain mit Blick auf den öffentlichen Sektor. Zuvor hat er Erfahrungen im Finanzsektor bei der Deutschen Bank gesammelt.
E-Mail: christof.tinnes@capgemini.com