9. Jänner 2017 Studie · Blockchain · Mobile Sicherheit · 18 Min. Lesezeit

Dezentralisierung zentralisierter Dienste

Archiviert. Dieser Inhalt wird nicht mehr aktualisiert. Er bleibt zur Information verfügbar.

Quelle https://technology2.a-sit.at/veroeffentlichungen/dezentralisierung-zentralisierter-dienste/

Über dieses Projekt wird demonstriert, wie Dienste, die traditionell auf zentralisierte Komponenten aufbauen, auch dezentral angeboten werden können. Zur Veranschaulichung wurde ein Prototyp zur Dezentralisierung von E-Mail erstellt.

Auf Grund der zunehmenden Verbreitung mobiler Geräte hat sich das NutzerInnen-Verhalten in den letzten Jahren stark verändert. Typischerweise haben Benutzerinnen und Benutzer mehrere Geräte, auf denen unterschiedliche Dienste genutzt werden, deren Daten möglichst überall verfügbar sein sollen. Dieses Verlangen wird von Serviceanbietern auch befriedigt, auf technischer Ebene allerdings meist durch den Einsatz klassischer Client-Server-Architekturen. Nach wie vor spielen zentrale Instanzen in verteilten Systemen dieser Art eine wichtige Rolle. Bestehende Dienste wurden über die Jahre hinweg zwar stetig weiterentwickelt und an die NutzerInnen-Gewohnheiten angepasst, ihre grundlegende Struktur bliebt jedoch weitgehend unverändert.

Im Rahmen dieses Projekts wurden Möglichkeiten untersucht, bestehende Dienste zu dezentralisieren und auf Endgeräte zu übertragen. Im Zuge dessen wurde ein Prototyp entwickelt, welcher einen dezentralen E-Mail-Service zur Verfügung stellt. Integrale Aspekte des umgesetzten Konzepts sind sichere Speicherung von Daten, transparente Entschlüsselung derselben bei Bedarf und Kompatibilität zu bestehenden Client-Anwendungen und externen Infrastrukturen. Außerdem wurde gezeigt, dass sich durch Transformation von zentralisierten Client-Server-Diensten in einen dezentralen Geräteverbund Vorteile im Bereich der Netzwerksicherheit ergeben können. Allerdings bleiben in diesem frühen Entwicklungsstadium noch einige diesbezügliche Fragen offen.

1. Einleitung

Die Art, wie internetbasierte Dienste genutzt werden, hat sich in den letzten Jahren durch den Vormarsch von Smartphones und Tabletcomputern radikal verändert. Immer seltener finden sich Nutzer und Nutzerinnen, welche nur über ein Endgerät verfügen, oder ihre Endgeräte völlig voneinander entkoppelt verwenden. Mit zunehmender Rechenleistung und wachsendem Speicherplatz auf elektronischen Geräten und hoher Verbreitung mobiler Endgeräte ergeben sich jedoch auch neue Möglichkeiten für die Umsetzung

dezentraler Dienste. Das etablierte Konzept der Synchronisation kann mit modernen Technologien auf eine vollständige Dezentralisierung erweitert werden. Zusätzlich muss ein solcher dezentralisierter Geräteverbund bei vernetzten Diensten wie E-Mail, Instant-Messaging und Web-Services nach außen hin transparent agieren. Dieses Ziel kann durch den Einsatz von Gateways erreicht werden. Somit ergibt sich die Möglichkeit der schrittweisen Integration bzw. der Ablöse bestehender, zentralisierter Dienste durch einen dezentralen Geräteverbund, ohne dass dies auf Kosten der Kompatibilität zu bestehender Infrastruktur geschieht. Durch den Einsatz eines Gateways, welcher keinen Zugriff auf einmal verteilte Daten hat, ergibt sich potenziell ein zusätzlicher Sicherheitsvorteil, da dessen Kompromittierung keine unmittelbaren Auswirkungen auf die im Geräteverbund gespeicherten Daten hat. Die Speicherung und Verarbeitung von Daten auf Endgeräten ohne zentrale Instanz wirft jedoch auch neue Sicherheitsfragen auf, die beantwortet werden müssen. Beispielsweise wird die Angriffsfläche durch die Verwendung mehrerer Geräte als Server auch vergrößert. Andererseits ergeben sich auch weitere Vorteile, da durch die replizierte, verteilte Speicherung von Daten innerhalb eines Geräteverbunds einige Risiken bezüglich Datensicherheit und Backups konzeptionell ausgeräumt werden können. Um es Nutzern und Nutzerinnen weiterhin zu ermöglichen, die ihnen bekannten Endbenutzerprogramme zum Zugriff auf Dienste zu verwenden, muss auch Kompatibilität zu Benutzerinnern und Benutzern hin gewährleistet sein. Die für die Umsetzung der eben beschriebenen Konzepte benötigten theoretischen Grundlagen werden im folgenden Abschnitt dargelegt. Anschließend wird anhand eines konkreten Beispiels die Praxistauglichkeit des Ansatzes demonstriert. Ausgehend davon wird nachfolgend die Architektur und Funktion des Prototyps zur Dezentralisierung von Diensten im Allgemeinen beschrieben. Abschließend wird ein Ausblick auf mögliche Weiterentwicklungen ausgehend von den vorgestellten Implementierungen gegeben.

2. Hintergrund

Um eine geräteübergreifende Nutzung von Diensten wie E-Mail, *Personal Information Management* oder verteiltes Arbeiten auf einem Datensatz zu gewährleisten, wurden traditionell bestehende Dienste und Anwendungen um Synchronisationsmechanismen erweitert und an geänderte Bedingungen angepasst, ohne diese jedoch grundlegend zu überarbeiten. Dadurch wurden diese Systeme über die Jahre hinweg zunehmen komplexer. Oft werden nach wie vor zentrale Instanzen eingesetzt, um Geräte miteinander zu vernetzen. Anbieter von Diensten haben dadurch in vielen Fällen Vollzugriff auf alle übertragenen Daten. Ein Beispiel hierfür ist der populäre Dateisynchronisationsdienst *Dropbox* [1] . Die Betreiber räumen ein, dass sie Zugriff auf alle Daten ihrer Kunden haben, um ihre Synchronisationsdienste anbieten zu können [1]. Betrachtet man jedoch die stetige Weiterentwicklung vor allem mobiler Endgeräte [2], stellt sich die Frage, ob eine derartig zentralisierte Vernetzung von Geräten noch zeitgemäß ist, obwohl aktuelle Geräte über ausreichend Ressourcen verfügen, um eine direkte Verteilung von Daten zu ermöglichen. Entwicklungen aus dem Bereich der *Peer-to-Peer-Netze* haben bewiesen, dass eine direkte Verbindung von Geräten und ein Informationsaustausch möglich ist, ohne dass zentrale Instanzen für die Orchestrierung diese Vorgänge benötigt werden. Bekannte Vertreter aus diesem Feld sind *CAN* [3], *Chord* [4], *Kademlia* [5] und *Pastry* [6]. Aktuell gewinnen jedoch vor allem Blockchain-basierte verteilte Applikationsplattformen wie *Ethereum* [7], *Nxt* [8], oder *Sia* [9] an Popularität. Allerdings sind solche konsensusbasierten Systeme, welche im Kern ohne Verfahren basierend auf *Proof-of-Work-* oder *Proof-of-Stake-* Mechanismen nicht korrekt arbeiten, an Einschränkungen gebunden und ungeeignet, beliebige bestehende Dienste zu dezentralisieren. Anonymisierungsnetzwerke wie z.B. *Tor* [10] stellen hingegen – wenn auch aus anderen Gründen – 1 https://dropbox.com/ Methoden zur Umsetzung direkt vernetzter Geräteverbunde zur Verfügung. Folglich sind auf Netzwerkebene alle notwendigen Voraussetzungen zur Transformation zentraler Dienste hin zu

verteilten Systemen gegeben. In Kombination mit leistungsfähiger Hardware und ausreichenden Bandbreiten sind somit die wichtigsten Voraussetzungen für die Dezentralisierung von Diensten unter realistischen Bedingungen erfüllt.

An der Grenze von Netzwerkebene zu Applikationsebene ergeben sich hingegen Probleme, welche gelöst werden müssen, um dezentralisierte Varianten bestehender Dienste zur Verfügung stellen zu können. Durch die Verlagerung von Diensten auf Endgeräte wird Benutzern und Benutzerinnen die Verantwortung für den Betrieb von Services auferlegt, welche mit der Außenwelt kommunizieren. Die Frage nach den Konsequenzen von kompromittierten Geräten innerhalb eines Verbunds muss daher geklärt werden, besonders da Nutzer und Nutzerinnen potentiell für den Missbrauch von Diensten auf ihren Endgeräten verantwortlich gemacht werden können. Dieses Projekt zielt jedoch primär darauf ab, die technischen Möglichkeiten zur Umsetzung eines Dezentralisierungsframeworks zu erforschen. Auf technischer Ebene gibt es Konzepte (und auch konkrete Umsetzungen), welche Fehlverhalten einzelner Knoten in einem Netzwerk identifizieren können. Die Palette reicht hierbei von grundlegenden Formalismen wie dem *Byzantine Generals Problem* [11] bis hin zu Systeme wie *PeerReview* [12] um anormales Verhalten zu detektieren und entsprechende Konsequenzen zu ziehen. Die Integration solcher Konzepte in ein Dezentralisierungsframework sprengt jedoch den Rahmen dieses Projekts und soll daher lediglich konzeptionell diskutiert werden. Es wurde stattdessen ein Prototyp entwickelt, welcher veranschaulicht, dass Dezentralisierung bestehender Dienste möglich ist und mit erhöhter Sicherheit einhergehen kann. Im nachfolgenden Abschnitt wird dies anhand eines Fallbeispiels illustriert und anschließend die Architektur des Frameworks zur Bereitstellung dezentraler Dienste allgemein beschrieben, welches sowohl nach außen hin Kompatibilität zu bestehender Infrastruktur gewährleistet, als auch Benutzern und Benutzerinnen gestattet, weiterhin die Client-Software zur Nutzung eines Dienstes zu verwenden, mit der sie vertraut sind.

3. Fallbeispiel E-Mail

Um zu demonstrieren, dass Dezentralisierung in der dargelegten Form praxistauglich ist, wurden die eingangs erwähnten Konzepte in Form eines Prototyps umgesetzt, welcher einen dezentralen E-Mail-Service zur Verfügung stellt. E-Mail wurde aus mehreren Gründen für eine Demonstration ausgewählt: Einerseits handelt es sich dabei um eine verteilte Anwendung, welche sowohl auf Smartphones, Tablets, Laptops und Desktop-Computern eingesetzt wird, wobei von mehreren Geräten aus auf derselben Datenbasis gearbeitet wird, welche auch überall verfügbar sein muss. Andererseits wurden bisher keine nennenswerten Versuche unternommen, das aktuelle serverbasierte System auch allumfassend zu dezentralisieren, ohne dass dies auf Kosten der Kompatibilität geschieht. Ein Beispiel für eine teilweise Umsetzung ist das von Hautakorpi et al. vorgestellte System [13], welches zwar Kompatibilität zu Benutzern hin anbietet, jedoch nicht in bestehende Infrastrukturen integrierbar ist. Außerdem bieten sich die eingesetzten Protokolle für die rasche Umsetzung eines Prototyps an, da diese (weitgehend) zustandslos sind. Nachdem die kleinsten Informationseinheiten innerhalb von Protokollen wie IMAPv4 [14] und SMTP [15] klar definiert sind, sind sowohl Replikation als auch Verteilung innerhalb eines Geräteverbunds vergleichsweise einfach umsetzbar.

Ein vorrangiges Ziel bei der Konzeption und Umsetzung einer dezentralen E-Mail-Implementierung war Abwärtskompatibilität sowohl gegenüber bestehender Infrastruktur als auch zum Endbenutzer hin. Hierfür wurde, ausgehend von einem SMTP-Server zum Senden und Empfangen von Nachrichten und eines IMAP-Servers zur Verteilung und Synchronisation von Nachrichten innerhalb eines Geräteverbunds, ein Framework geschaffen. Dadurch, dass auf bestehenden, protokollkonformen Implementierungen

aufgebaut wurde, und deren öffentliche Schnittstellen unberührt blieben, konnte Abwärtskompatibilität sichergestellt werden. Des Weiteren gibt es historisch bedingt eine klare Trennung zwischen *Mail Transport Agent* (MTA) zum Senden und Empfangen und *Mail Delivery Agent* (MDA) zum Zustellen von Nachrichten in die Postfächer von Nutzern und Nutzerinnen. Ein dezentraler E-Mail Dienst lässt sich somit umsetzen, indem ein MDA eingeführt wird, welcher eingehende Nachrichten an den *Event Transformation and Distribution Core* (ETDC, siehe Abschnitt 4) übermittelt, welcher diese innerhalb eines Geräteverbunds verteilt. Die Geräte eines Verbunds sind im einfachsten Fall vor der ersten Benutzung vom Benutzer bzw. der Benutzerin zu definieren und in Konfigurationsdateien einzutragen.

Abbildung 1: Funktionsweise eines E-Mail-Systems In einem weiteren Schritt wurde der für Dezentralisierung entwickelte MDA dahingehend erweitert, dass auch vom E-Mail-Client, (dem *Mail User Agent* (MUA)) ausgehende IMAP-an den ETDC weitergeleitet und auf allen Geräten im Verbund repliziert werden. Dadurch ergibt sich die in Abbildung 2 dargestellte Struktur: Auf jedem Gerät werden SMTP- und IMAP-Server betrieben. Benutzer und Benutzerinnen konfigurieren ihre E-Mail-Clients so, dass diese sich zu den lokal betrieben Servern verbinden. Somit fungieren die eingesetzten SMTP- und IMAP-Server als lokale Proxies. Zusammengefasst geht die Dezentralisierung durch den Einsatz angepasster Varianten von MTAs und MDAs für Nutzer und Nutzerinnen vollkommen transparent vonstatten.

Nachdem die bestehende globale E-Mail-Infrastruktur generell stark vom Domain Name System abhängig ist, treffen auf das Empfangen von Nachrichten ähnliche Einschränkungen zu wie auf das Versenden. Durch die flexible Architektur des implementierten Prototyps besteht jedoch die Möglichkeit, jedes Gerät innerhalb eines Verbunds unterschiedlich zu konfigurieren. Tatsächlich kann ein reiner Gateway-Knoten betrieben werden, der ausschließlich für die Kommunikation mit der Außenwelt verantwortlich ist und selbst keine Daten speichert, sondern diese lediglich weiterleitet.

Um ein zusätzliches Maß an Sicherheit zu bieten, werden alle eingehenden Nachrichten verschlüsselt, bevor diese abgespeichert werden. Die Entschlüsselung erfolgt transparent, sobald ein Nutzer, bzw. eine Nutzerin zuvor gespeicherte Nachrichten abrufen möchte. Dies wird durch den

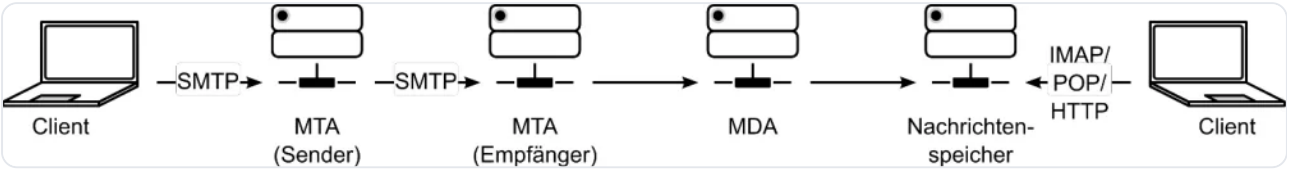
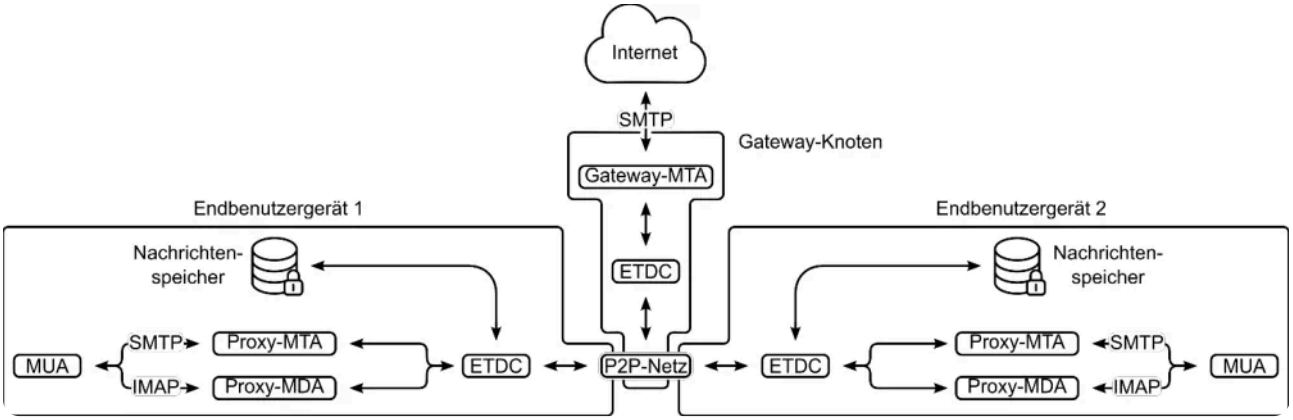


Abbildung 2: Dezentrales E-Mail-System mit Gateway-Knoten Tatsächlich gibt es jedoch im E-Mail-Anwendungsfall eine Einschränkung, welche vollständige Dezentralisierung unter realistischen Bedingungen verhindert: Auf Grund der allgemeinen Spam-Problematik werden E-Mails, welche von Servern ohne gültigen DNS-Eintrag versendet werden, nicht zugestellt [16]. Daher muss zumindest der MTA auf einem im Domain Name System registrierten Server betrieben werden, um auch mit der Außenwelt kommunizieren zu können. Um diesem Umstand gerecht zu werden, wurde auch der MTA dahingehend angepasst, dass Nachrichten nicht direkt vom lokalen SMTP-Server versendet werden, sondern diese zuerst an einen definierten Knoten innerhalb des Verbunds weitergeleitet und von dort aus versendet werden. Ein wesentlicher Unterschied zum Status Quo ist, dass für diese Übertragung der MUA und der versendende SMTP-Server nicht direkt miteinander kommunizieren, sondern wie in Abbildung 2 dargestellt die bestehende Verbindung zwischen den Knoten im Geräteverbund verwendet wird.



Einsatz asymmetrischer Kryptografie in Kombination mit den im Zusammenhang mit E-Mail üblichen Authentifizierungsverfahren ermöglicht. Im nachfolgenden Abschnitt wird näher auf die Konzepte im Allgemeinen eingegangen, die in diesem Fall konkret beschrieben wurden und die Architektur des Dezentralisierungsframeworks beschrieben.

4. Architektur

Der im Rahmen dieses Projekts entwickelte Prototyp zu Dezentralisierung zentralisierter Dienste besteht konzeptionell aus fünf Komponenten:

- Gateway zur Kommunikation nach außen hin und zur transparenten Integration mit bestehenden, externen Systemen
- Peer-to-Peer Netzwerklayer zur sicheren Übertragung von Informationen innerhalb eines Geräteverbunds
- Persistenter Speicher in Kombination mit einem Krypto-Layer zur automatischen Verschlüsslung eingehender Daten und transparenter Entschlüsselung bei Bedarf
- Lokale Proxies zur transparenten Integration mit bestehenden Client-Applikationen
- *Event Transformation and Distribution Core* (ETDC) für die Umwandlung Servicespezifischer Daten in anwendungsunabhängige *Events* welche auf Knoten im Netzwerk repliziert werden

Aus dieser modularen Architektur ergibt sich einerseits die Möglichkeit, einzelne Komponenten auszutauschen, vor allem werden dadurch jedoch die Voraussetzungen geschaffen, Knoten eines Geräteverbunds unterschiedlich zu konfigurieren. Wie bereits im E-Mail-Fallbeispiel angedeutet, bietet es sich an, Knoten, welche über ein Gateway mit der Außenwelt verbunden sind, ohne persistenten Speicher zu betreiben. Generell gilt, dass reine Gateway-Knoten lediglich eine Schnittstelle zur Außenwelt zur Verfügung stellen, welche eingehende Daten innerhalb des Geräteverbunds verteilt und ausgehende Daten weiterleitet. Des Weiteren kann ein Gateway-Knoten so konfiguriert werden, dass dieser keine Daten von anderen Geräten des Verbunds abrufen kann. Nachdem sich an einem Gateway eine besonders große Angriffsfläche ergibt, kann durch den Wegfall eines Speichers der Schaden im Fall erfolgreicher Angriffe auf einen solchen Knoten minimiert werden: Da keine Daten lokal gespeichert, sondern im Verbund verteilt werden, können bestehende Daten auch nicht ausgelesen und/oder manipuliert werden. Dies ist ein klarer Vorteil, verglichen mit bestehenden serverzentrischen Systemen, welche alle Daten auch an dem Ort

lagern, an dem eine Verbindung nach außen besteht. Abbildung 3 illustriert ein Netzwerk bestehend aus einem reinen Gateway-Knoten und zwei weiteren Knoten ohne Gateway-Funktionalität. Durch diese redundante Speicherung von Daten ergeben sich auch Möglichkeiten, Backupfunktionalitäten vergleichsweise einfach umzusetzen.

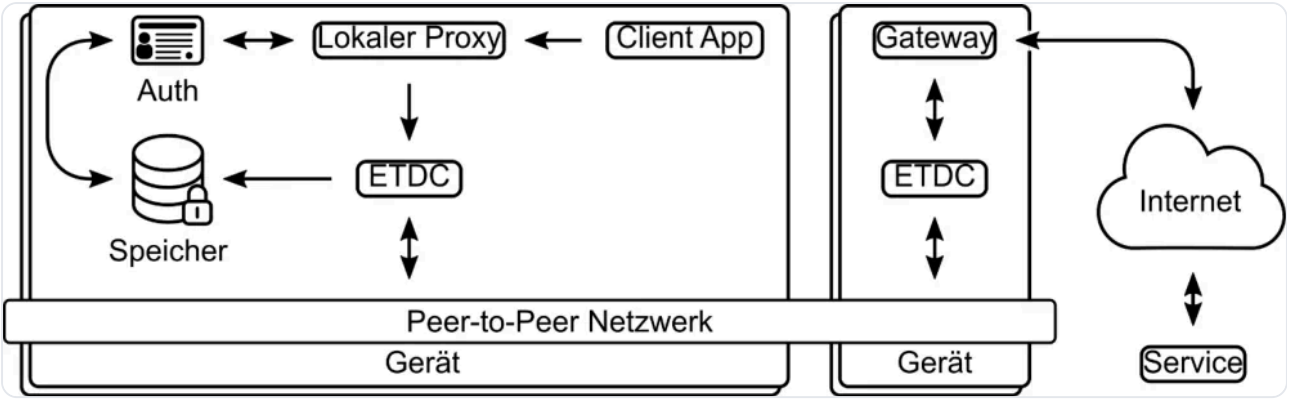


Abbildung 3: Struktur eines Netzwerks mit drei Knoten Ein weiterer Sicherheitsgewinn kann durch den Einsatz eines modernen Netzwerklayers in Kombination mit lokalen Proxies erzielt werden: Da alle Geräte innerhalb eines Verbunds unter der Kontrolle eines einzelnen Nutzers bzw. einer einzelnen Nutzerin stehen, und Clientsoftware nur mehr mit lokalen Proxies verbunden wird, muss keine Rücksicht auf eventuell veraltete

Endbenutzersoftware genommen werden. Folglich können zeitgemäße Methoden zur Absicherung aller Verbindungen zwischen den Geräten eingesetzt werden, um Angriffe auf Netzwerkebene effektiver abwenden zu können. So kann von vornherein ausgeschlossen werden, dass unsichere Verschlüsselungsverfahren verwendet werden und *Perfect Forward Secrecy* kann durch den Einsatz moderner kryptografischer Methoden garantiert werden. Dies steht im klaren Gegensatz zur aktuellen Situation von internetbasierten Diensten, welche auch teilweise unsichere Verfahren anbieten (müssen), um Services einer möglichst breiten Zielgruppe anbieten zu können.

Im Folgenden werden die Funktionalität des Dezentralisierungsframeworks, das Empfangen, Verteilen und die Speicherung eingehender Daten genauer beschrieben.

4.1. Verarbeitung eingehender Daten

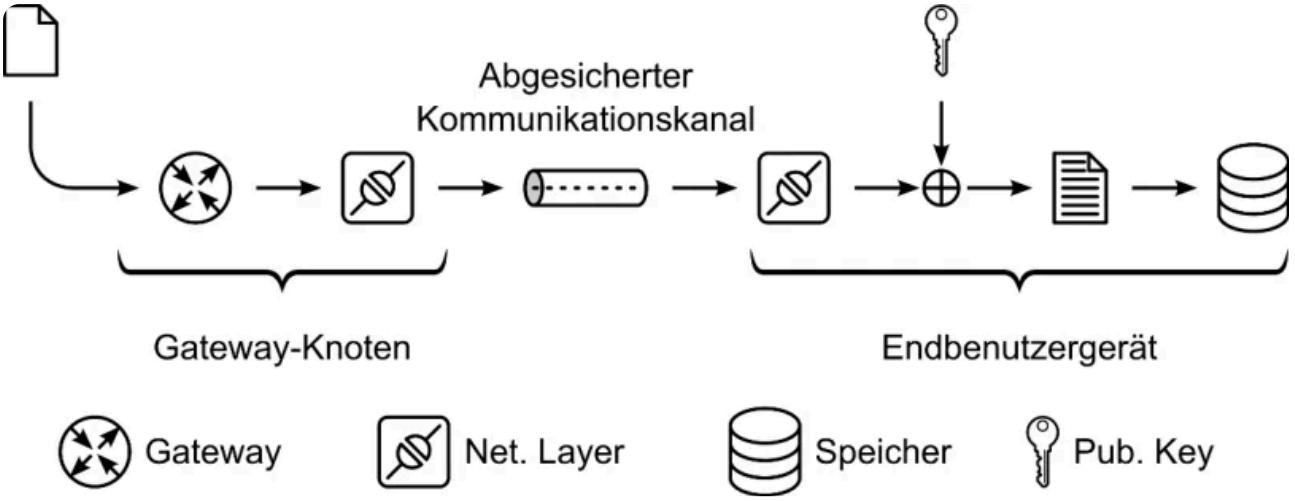
Treffen Daten am Gateway ein, werden diese (im Fall eines reinen Gateway-Knotens) über den Netzwerk-Layer an alle anderen Geräte im Verbund übertragen. Der hierfür notwendige sichere Kommunikationskanal wird aktuell über Tor hergestellt. Jeder Knoten ist als *Hidden Service* im Tor-Netz erreichbar, und kann auch nur kontaktiert werden, wenn der *Hidden Service Identifier* bekannt ist. Tor erlaubt es Hidden Services außerdem, nur Verbindungen autorisierter Knoten anzunehmen. Somit kann sichergestellt werden, dass nur Knoten innerhalb eines Verbunds miteinander kommunizieren können. Des Weiteren garantiert Tor durch den Einsatz moderner kryptografischer Verfahren Perfect Forward Secrecy. Da Netzwerksicherheit und verschlüsseltes Speichern von Daten somit voneinander entkoppelt sind, ist es nicht notwendig, Schlüsselmaterial innerhalb des Geräteverbunds auszutauschen. Stattdessen verfügt jeder Knoten über eigene Schlüssel und Daten werden vor der Übertragung nicht explizit verschlüsselt, da Transportverschlüsselung Aufgabe des Netzwerklayers ist. Jeder Knoten, welcher Daten auch persistent speichern soll, benötig daher lediglich ein *public/private key pair* .

Abbildung 4: Verarbeitung eingehender Daten Sobald ein Knoten Daten empfängt, werden diese unter dem öffentlichen Schlüssel dieses Knotens verschlüsselt und abgespeichert. Abbildung 4 veranschaulicht den gesamten Prozess ausgehend von Daten, welche von außerhalb in einem Geräteverbund abgespeichert werden sollen.

Ein Vorteil, welcher sich durch diese späte Verschlüsselung am Gerät ergibt, ist, dass z.B. Metadaten oder gezielt andere Informationen ausgelesen werden können und der Nutzer bzw. die Nutzerin über neue eingehende Daten informiert werden kann. Sobald auf die Daten zugegriffen wird, werden diese transparent entschlüsselt. Dies wird einerseits durch den Einsatz asymmetrischer Kryptografie ermöglicht, andererseits auch durch eine Eigenschaft nahezu aller Online-Dienste: Typischerweise müssen sich Nutzer und Nutzerinnen authentifizieren, um Services zu nutzen. Ausgehend von den Authentifizierungsinformationen kann transparent ein Schlüssel für die Entschlüsselung gespeicherter Daten abgeleitet werden. Nähere Details hierzu werden im folgenden Abschnitt behandelt.

4.2. Transparente Entschlüsselung gespeicherter Daten

Ein Ziel bezüglich sicher Speicherung von Daten ist, dass diese immer verschlüsselt abgelegt werden, und erst bei Bedarf entschlüsselt werden. Ausgehend von der Tatsache, dass bestehende



Internet-Dienste dezentralisiert werden sollen, kann auf eine Eigenschaft zurückgegriffen werden, welche auf viele Dienste zutrifft: Will man einen Dienst nutzen, muss man sich authentifizieren. Aus einer Vielzahl von Gründen wird in der Regel auf passwortbasierte Authentifizierung zurückgegriffen. Dienst und Nutzer teilen sich folglich geheimes Wissen. Dieser Umstand kann in Kombination mit asymmetrischer Kryptografie herangezogen werden um verschlüsselte Speicherung eingehender Daten und transparente Entschlüsselung bei Bedarf umzusetzen. Im Rahmen dieses Projekt wurde dafür wie folgt vorgegangen:

- An jedem Knoten wird ein public/private key pair generiert.
- Der öffentliche Schlüssel wird verwendet, um eingehende Daten zu verschlüsseln.
- Der private Schlüssel wird mit einem von der Authentifizierungsinformation abgeleiteten Schlüssel verschlüsselt abgespeichert. Der so geschützte private Schlüssel wird als *Wrapped Key* bezeichnet.

- Sobald ein Nutzer oder eine Nutzerin den vom Dezentralisierungsframework angebotenen Dienst nutzt und sich dafür authentifiziert, wird der private Schlüssel entschlüsselt und damit die zuvor gespeicherten Daten dechiffriert.

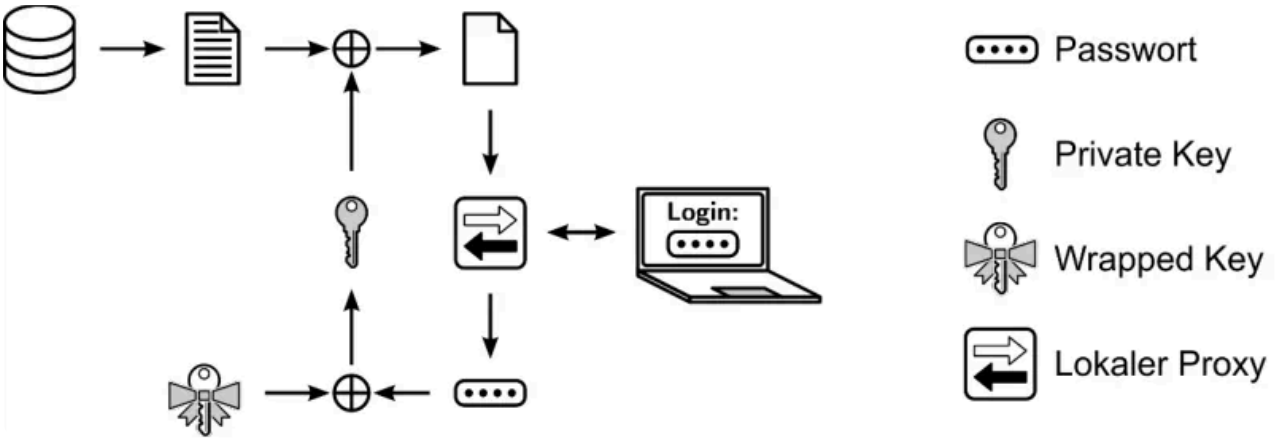
Ein weiterer Vorteil aus diesem Konzept ist, dass weder ein Passwort, noch der Hash eines Passworts gespeichert werden müssen, und trotzdem passwortbasierte Authentifizierung umgesetzt werden kann. Die Überprüfung der Authentifizierungsinformation besteht schlichtweg darin, den privaten Schlüssel zu entschlüsseln. Gelingt dies, wurden die korrekten Anmeldedaten eingegeben. Wenn nicht, schlägt der Authentifizierungsversuch fehl. Um Brute-Force-Angriffe zu unterbinden, wird eine Schlüsselableitungsfunktion eingesetzt, welche *key stretching* in entsprechendem Ausmaß garantiert. Abbildung 5 veranschaulicht den eben beschriebenen Prozess der transparenten Entschlüsselung von Daten.

Abbildung 5:Transparente Entschlüsselung von Daten

5. Ausblick

Die prinzipielle Machbarkeit eines Frameworks für die Dezentralisierung von Diensten wurde erfolgreich demonstriert. Bei dem umgesetzten Prototyp [2] handelt es sich jedoch um einen auf einen konkreten Anwendungsfall spezialisierten Demonstrator. Verfahren zum Abrufen von neuen Informationen aus dem Geräteverbund nach längeren Offline-Phasen wurden nicht praktisch behandelt. Ebenso wurden Mechanismen, welche die Speicherplatzheterogenität unterschiedlicher Geräte innerhalb eines Verbunds berücksichtigen, nicht implementiert, da dies mit der Möglichkeit, gezielt Informationen abzurufen, Hand in Hand geht. Konzeptionell gibt es unterschiedliche Herangehensweisen, um diese Probleme zu lösen. Ein entscheidender Aspekt dabei ist der Umgang mit kompromittierten Geräten innerhalb des Verbunds: Wenn (wie beim aktuellen Prototyp) keine Möglichkeit besteht, aktiv Informationen anzufragen, sondern diese nur an andere Teilnehmer gesendet werden können, fällt ein kritischer Angriffsvektor eines derart dezentralisierten Systems weg. Zu keiner Zeit kann ein kompromittiertes Gerät an nicht lokal gespeicherte Informationen gelangen, welche ihm nicht von anderen Teilnehmern selbständig übermittelt werden. Es ist somit nicht möglich, beliebige Daten aus dem Netzwerk abzugreifen.

2 https://demo.a-sit.at/wp-content/uploads/2017/01/decentral_demo.zip



Gleichzeitig kann jedoch auch die Speicherplatzheterogenität nicht dahingehend berücksichtigt werden, dass einmal lokal gelöschte Daten nachträglich abgerufen werden, sollte dies von Nutzern und Nutzerinnen gewünscht sein. Klarerweise können auch Offline-Phasen somit nicht kompensiert werden. Die Erkennung von Fehlverhalten innerhalb eines verteilten Systems wurde bereits vielfach behandelt. Allerdings sind die Voraussetzungen im Kontext eines dezentralisierten Service für den Privatgebrauch andere als im Unternehmenskontext. Allein durch die typischerweise unterschiedliche Anzahl von Geräten innerhalb eines solchen Verbunds ergeben sich radikal andere Rahmenbedingungen: Können Ausprägungen byzantinischer Fehler, wie sie im Fall des vorgestellten Dezentralisierungsansatzes zu erwarten sind, innerhalb eines größeren Netzwerks noch zufriedenstellend gelöst werden, ist dies innerhalb eines kleinen Verbunds von drei Geräten nur schwer möglich. Gerade dieser Grenzfall kann jedoch nicht vernachlässigt werden, wenn Nutzungsgewohnheiten im privaten Kontext berücksichtigt werden sollen. Hier müssen andere Mechanismen und Protokolle zum Einsatz kommen, um mit kompromittierten Knoten eines dezentralen Netzwerks umzugehen. Die Frage nach dem Aufwand der betrieben werden muss, damit Endbenutzergeräte wie Smartphones erfolgreich attackiert werden können, und Daten aus dem Geräteverbund zu extrahieren, lässt sich im Vergleich zu traditionellen Serversystemen nicht ohne Weiteres beantworten. Im Speziellen ist jedoch die Architektur der verwendeten Betriebssysteme zu beachten, welche von sich aus effektive Sandboxing-Mechanismen erzwingt und eine starke Isolation zwischen einzelnen Prozessen und Applikationen garantiert – selbst Endbenutzern und Endbenutzerinnen ist es nicht möglich, an Daten zu gelangen, welche ihnen nicht „freiwillig“ von einer Applikation zugänglich gemacht werden. Grund dafür ist, das typische Smartphone-Betriebssysteme ihren Nutzern und Nutzerinnen schlichtweg nur sehr selektiv und eingeschränkt Zugriff auf das Dateisystem erlauben. Unter bestimmten Voraussetzungen lassen sich einige Gefahren somit auf ein Restrisiko reduzieren. In diesem Zusammenhang muss jedoch auch die Frage beantwortet werden, ob Nutzer und Nutzerinnen selbst potentielles Fehlverhalten eines Geräts erkennen können. Kann dies bejaht werden, lassen sich Abwehrmaßnahmen konzipieren, welche den Benutzer, bzw. die Benutzerin miteinbeziehen. Im anderen Fall gibt es zumindest die Möglichkeit auf Systeme wie das eingangs erwähnte PeerReview zurückzugreifen, und den Nutzer oder die Nutzerin bereits bei vermutetem Fehlverhalten zu informieren. Nachdem alle Geräte unter der Kontrolle einer einzigen Person stehen, kann dies ausreichend sein, um größeren Schaden abzuwenden – im Zweifelsfall kann durch manuelle Intervention ein Gerät vorübergehend aus dem Verbund entfernt werden, bis Klarheit geschaffen wurde.

6. Referenzen

[1] Dropbox, Inc, „Dropbox Privacy Policy,“ 16 02 2016. [Online]. Available:

https://www.dropbox.com/privacy/. [Zugriff am 27 09 2016].

[2] Primate Labs Inc., „MacBook Pro (13-inch Retina Late 2013) vs iPhone 7,“ 2016. [Online].

Available: https://browser.primatelabs.com/v4/cpu/compare/545914?baseline=549149.

[Zugriff am 26 09 2016].

[3] S. Ratnasamy, P. Francis, M. Handley und R. S. Karp, „A Scalable Content-addressable Network,“ in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, San Diego, ACM, 2001, pp. 161-172.

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek und H. Balakrishnan, „Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, San Diego, ACM, 2001, pp. 149-160.

[5] P. Maymounkov und D. Mazières, „Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, Springer, 2002, pp. 53-65.

[6] A. Rowstron und P. Druschel, „Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Middleware 2001*, Springer, 2001, pp. 329-350.

[7] Ethereum Foundation, „Ethereum," [Online]. Available: https://www.ethereum.org/. [Zugriff am 07 12 2016].

[8] „Nxt - The Blockchain Application Platform," [Online]. Available: https://nxt.org/. [Zugriff am 11 11 2016].

[9] Nebulous Inc., „Sia," [Online]. Available: https://sia.tech/. [Zugriff am 11 11 2016].

[10] R. Dingledine, N. Mathewson und P. Syverson, „Tor: The Second-Generation Onion Router,"

in *Proceedings of the 13th USENIX Security Symposium*, San Diego, 2004.

[11] L. Lamport, R. Shostak und M. Pease, „The Byzantine generals problem," in *ACM Transactions on Programming Languages and Systems (TOPLAS)*, ACM, 1982, pp. 382-401.

[12] A. a. K. P. a. D. P. Haeberlen, „PeerReview: Practical Accountability for Distributed Systems,"

in *Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles*, Stevenson, Washington, ACM, 2007, pp. 175-188.

[13] J. Hautakorpi, G. Camarillo und D. Lopez, „Framework for Decentralizing Legacy Applications," in *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, IEEE Computer Society, 2009, pp. 544-549.

[14] M. R. Crispin, „INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1," 03 2003.

[Online]. Available: https://tools.ietf.org/html/rfc3501. [Zugriff am 18 08 2016].

[15] J. B. Postel, „SIMPLE MAIL TRANSFER PROTOCOL," 08 1982. [Online]. Available:

https://tools.ietf.org/html/rfc821. [Zugriff am 07 12 2016].

[16] S. Kitterman, „Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1," 04 2012. [Online]. Available: https://tools.ietf.org/html/rfc7208. [Zugriff am 07 12 2016].