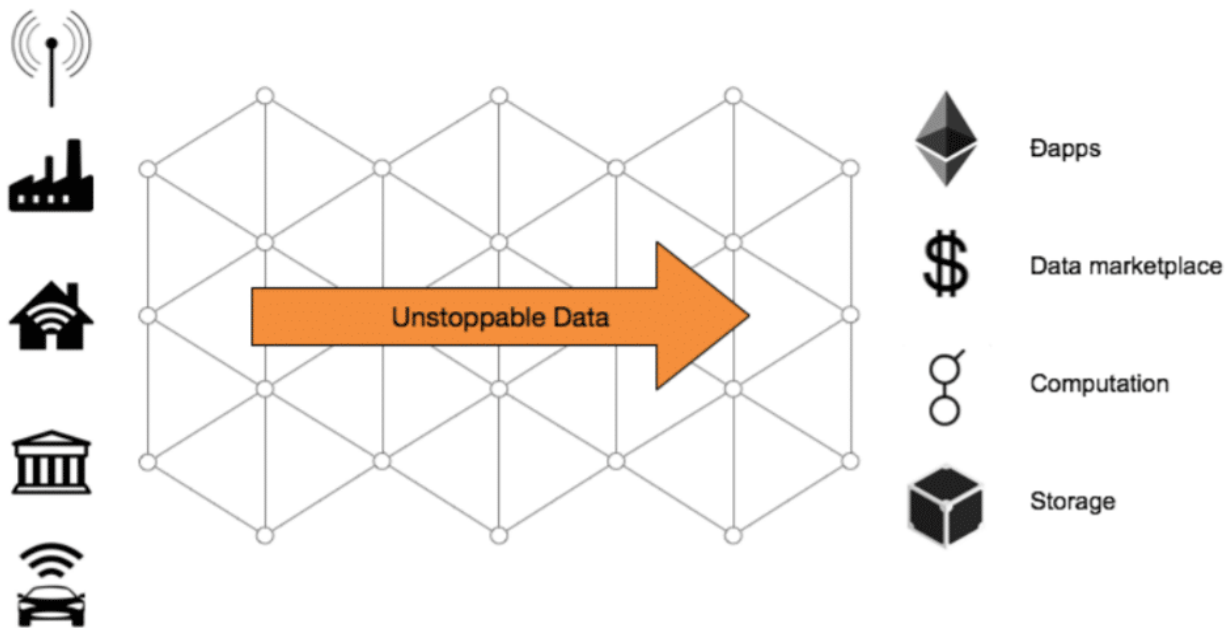


Unaufhaltsame Daten für unaufhaltsame Anwendungen: DATAcoin von Streamr

**streamr**

25. Juli 2017  
Version 1.0

Dieses Whitepaper dient nur zu Informationszwecken und stellt weder ein Angebot noch eine Anlageempfehlung oder eine Empfehlung zur Investition dar. Jedes Element dieses Whitepapers kann im weiteren Verlauf des Projekts signifikanten Änderungen unterzogen werden.



## Streamr-Vision

Streamr liefert unaufhaltsame Daten an unaufhaltsame Anwendungen. Es ist die Echtzeit-Datenwirbelsäule des globalen Supercomputers. Es handelt sich um ein dezentrales Netzwerk für eine skalierbare Datenübermittlung mit niedrigen Latenzzeiten und einem Schutz gegen Eingriffe von außen, das mit dem DATAcoin-Token betrieben wird. Jedermann - oder jede Maschine - kann neue Daten in sogenannten Datenströmen (Datastreams) veröffentlichen und andere können diese Datenströme abonnieren, um damit Dapps, intelligente Verträge, Mikrodienstleistungen (Microservices) und intelligente Datenpipelines zu betreiben.

Um die Teilnahme der Benutzer am Netzwerk zu fördern, gibt es einen eingebauten Anreizmechanismus für die Monetarisierung der Daten.

Wertvolle Daten von Börsen, angeschlossenen Geräten, IoT-Sensoren und Social-Media-Kanälen können Unternehmen, Entwicklern und Privatpersonen angeboten werden. Maschinen können ihre Daten autonom verkaufen, werden bezahlt und kaufen die Daten, die sie benötigen. So entsteht ein globaler Markt für Echtzeitdaten mit integrierter Datenprovenienz, Verschlüsselung und Zugriffskontrollen.

Neben dem dezentralen Datennetzwerk und dem Marktplatz umfasst der vollständige Technologiebaum von Streamr (Streamr Stack) eine leistungsstarke Analyse-Engine und ein UI für die schnelle Entwicklung von Echtzeit-Dapps. Datenströme, intelligente Verträge sowie dezentrale Ressourcen für Rechenleistung können in einer Low-Code-Umgebung mit Hilfe von High-Level-Bausteinen miteinander verbunden werden. Streamr wird die einfachste Möglichkeit für die Erstellung von datengesteuerten und vertrauenswürdigen Blockkettenanwendungen in Echtzeit sein.

Eine Revolution vollzieht sich gerade dort, wo zentralisierte Cloud-Dienste einer nach dem anderen durch tokenisierte, dezentrale Lösungen ersetzt werden. Golem zum Beispiel ersetzt die Azure Virtual Machine und IPFS tritt an die Stelle von Azure Blob Storage. Streamr ist stolz darauf, sich dieser Revolution anzuschließen, indem es eine dezentrale Lösung für das Messaging und die Eventverarbeitung anbietet und damit Plattformen wie Azure EventHub und Azure Stream Analytics ersetzt

## 1. Hintergrund

Echtzeitdaten werden in den nächsten Jahren immer mehr zur Ware. Riesige Mengen an zeitgestempelten Daten werden durch Sensoren und angeschlossene Geräte in der Fertigung, dem Dienstleistungssektor und der gesamten Lieferkette erzeugt - also all jenen Bereichen, die der modernen Wirtschaft zugrunde liegen. Ein großer Teil dieser Daten wird im Streaming-Verfahren erzeugt<sup>1,2</sup>

Die Datenmenge steigt exponentiell mit dem Wachstum von IoT und der Allgegenwart von miteinander verbundenen Geräten. Für den globalen IoT-Markt wird durch IHS Markit<sup>3</sup> prognostiziert, dass die Menge von 15,4 Milliarden Geräten im Jahr 2015 auf 30,7 Milliarden Geräte im Jahr 2020 und auf 75,4 Milliarden Geräte im Jahr 2025 anwachsen wird. Ein Großteil der neu generierten Daten ist wertvoll: sie können zur Optimierung von Fertigungsprozessen genutzt werden oder um den Zustand und den Reiseweg von Gütern mit hoher Präzision zu verfolgen. Ferner können mit diesen Daten maßgeschneiderte Lösungen für bestehende Verbraucherdienste mit hoher Qualität erschaffen werden oder es werden gänzlich neue Dienstleistungen und Geschäftsmodelle kreiert.

Gleichzeitig entsteht momentan ein Megatrend im Hinblick auf die nächste Generation der Computer. In einer Zukunft, die auf verteilten Verzeichnissen basiert, läuft der Backend-Code von dezentralen Anwendungen - oder Dapps<sup>4</sup>- in Peer-to-Peer-Netzwerken. Ethereum ist bereits eine solche Dapp, ebenso wie Golem; und es befinden sich noch viele mehr in der Entwicklung.

Allerdings laufen Dapps nicht isoliert: Sie benötigen externe Daten, um zu funktionieren. Derzeit ist die Lagerung und die Verteilung von realen Daten zentralisiert, wodurch Dapps anfällig für all die bekannten Probleme wie Machtballung, mangelnde Robustheit und Anfälligkeit für Cyberangriffe sind.

Sie können heute bereits Daten auf der Blockkette speichern. Es gibt auch dezentrale Dateispeicheranwendungen wie IPFS, Swarm und Storj und auch Datenbanken wie BigchainDB beginnen sich zu entwickeln.

Solche Lösungen sind zwar sicherlich Teil der neuen dezentralen Struktur, aber sie sind nicht wirklich eine Lösung für Fälle, in denen Echtzeitdaten in signifikanten Mengen benötigt werden. Die Blockkette ist nicht für einen hohen Durchsatz oder niedrige Latenzen ausgelegt, sie skaliert nicht, und der Speicherplatz ist teuer.

Benötigt wird eine native, dezentrale Datenwirbelsäule als Ergänzung zu dezentralisierten Apps. Diese Echtzeit-Datenwirbelsäule ist das derzeit noch fehlende Bindeglied, zu dessen Erschaffung wir beitragen wollen. Die von uns erschaffene Infrastruktur besteht aus einer Sammlung von Technologien, die uns bei der Verbindung von Computern in einem globalen Peer-to-Peer-Netzwerk (P2P) helfen und gleichzeitig Anreize zur Teilnahme weiterer Mitglieder schaffen soll. Dies ist ein

---

<sup>1</sup>Susan O'Brien: "5 Big Data Trends Shaping the Future of Data-Driven Businesses", Datameer, 11. Mai 2016 (<https://www.datameer.com/company/datameer-blog/5-big-data-trends-shaping-future-data-driven-businesses/>)

<sup>2</sup>Tony Baer: "2017 Trends to Watch: Big Data", Ovum, 21. November 2016. ([https://ovum.informa.com/~media/Informa-Shop-Window/TMT/Files/Whitepapers/2017\\_Trends\\_to\\_Watch\\_Big\\_Data.pdf](https://ovum.informa.com/~media/Informa-Shop-Window/TMT/Files/Whitepapers/2017_Trends_to_Watch_Big_Data.pdf))

<sup>3</sup>Sam Lucero: "IoT Platforms: enabling the Internet of Things", IHS Markit, März 2016. (<https://cdn.ihs.com/www/pdf/enabling-IOT.pdf>)

<sup>4</sup>Für eine (englischsprachige) Definition des Begriffs „Dapps“, siehe „Johnston et al.: The General Theory of Decentralized Applications“ (<https://github.com/DavidJohnstonCEO/DecentralizedApplications>)

Netzwerk, das niedrige Latenzzeiten, eine robuste und sichere Datenbereitstellung und Persistenz bietet, und das alles in einem jeweils passenden Maßstab. Dapps werden in der Zukunft von Daten betrieben, und unsere Mission ist es dafür zu sorgen, dass die Daten auch ungehindert fließen können.

Wir erschaffen auch einen Markt für Echtzeitdaten. Auf dem Datenmarktplatz kann jedermann Veranstaltungen in Datenströmen (Datastreams) veröffentlichen, und jeder kann Streams abonnieren und die Daten in dezentralen Anwendungen nutzen. Ein Großteil der Daten ist kostenlos. Wenn das nicht der Fall ist, werden die Nutzungsbedingungen in intelligenten Verträgen (Smart Contracts) auf Ethereumbasis gespeichert. Ein digitales Token - ein DATAcoin - wird benötigt, um auf den Marktplatz zuzugreifen, Streams abzurufen sowie zur Kompensation von Knotenpunkten im P2P-Netzwerk. Abonnenten zahlen für die Daten mit den Token, während Datenproduzenten und Netzwerkteilnehmer auf gleichem Wege automatisch und sicher für ihren Aufwand entlohnt werden.

Unsere Technologie baut auf einer dezentralen Transportschicht auf. Neben einer größeren Robustheit und Belastbarkeit ermöglicht die Dezentralisierung auch eine geringere Fehlertoleranz, mehr Offenheit, Transparenz und die Bildung einer Gemeinschaft.

Die Macht über Daten liegt damit nicht mehr bei großen Unternehmen wie Google, Amazon, Microsoft und IBM. Das Netzwerk besteht aus einer Vielzahl von Datenproduzenten, Datenkonsumenten und Message-Broker-Knoten. Sie erarbeiten sich einen guten Ruf für sich selbst und verdienen wertvolles Karma, indem Sie einen Beitrag zum Datenaustausch und für den Betrieb des Netzwerks zum Wohle aller leisten.

Wir glauben, dass ein nachhaltiges Wachstum der Blockkettengemeinschaft durch die Schaffung einer guten Anwenderoberfläche gefördert wird. Werkzeuge werden benötigt, damit auch Laien sichere, intelligente Verträge entwickeln können und diese Verträge und Dapps mit zuverlässigen Datenquellen verbinden können. Wir helfen beim Erschaffen der benötigten Werkzeuge mit einem visuellen Editor, Wrappern und Vorlagen. Kurz gesagt: wir wollen der ideale Anlaufpunkt für alle werden, die sich mit der Erstellung datengetriebener, dezentraler Dienste beschäftigen.

Im weiteren Verlauf dieses Dokuments beschreiben wir den Streamr-Technologiebaum (Streamr Stack), definieren die Rolle der digitalen Token, erläutern den Status Quo und stellen Ihnen die F&E-Roadmap sowie auch unser Team vor.

## 2. Streamr-Technologiebaum

Die dezentrale Echtzeit-Datenpipeline baut auf einem mehrschichtigen Technologiebaum auf:

- **Streamr Editor** stellt eine Usability-Ebene und eine Werkzeugsammlung dar, der eine schnelle Entwicklung von dezentralen, datengesteuerten Anwendungen ermöglicht.
- **Streamr Engine** ist eine hochleistungsfähige Eventverarbeitungs- und Analyse-Engine, die unabhängig von der Blockkette ausgeführt wird. Sie kann auf einem Anbieter für dezentrale Rechnerleistung - wie Golem - laufen.
- **Streamr Data Marketplace** ist ein Universum von gemeinsam genutzten Datenströmen, zu denen jeder beitragen und die jeder abonnieren kann.
- **Streamr Network** ist die Datentransportschicht, die den Rahmen für ein anreizbasiertes Peer-to-Peer Netzwerk zu Messagingzwecken innerhalb der dezentralen Datenpipeline

definiert.

- **Streamr Smart Contracts** ermöglichen es den Knoten im Streamr-Netzwerk, einen Konsens zu erzielen, Stream-Metadaten zu speichern, Genehmigungen und Integritätsprüfungen zu verwalten und sie dienen auch der Sicherheit bei Tokentransfers.

Der folgende Abschnitt geht jede Schicht des Technologiebaums (siehe Abbildung 1) im Detail durch und folgt einem Top-Down-Ansatz.

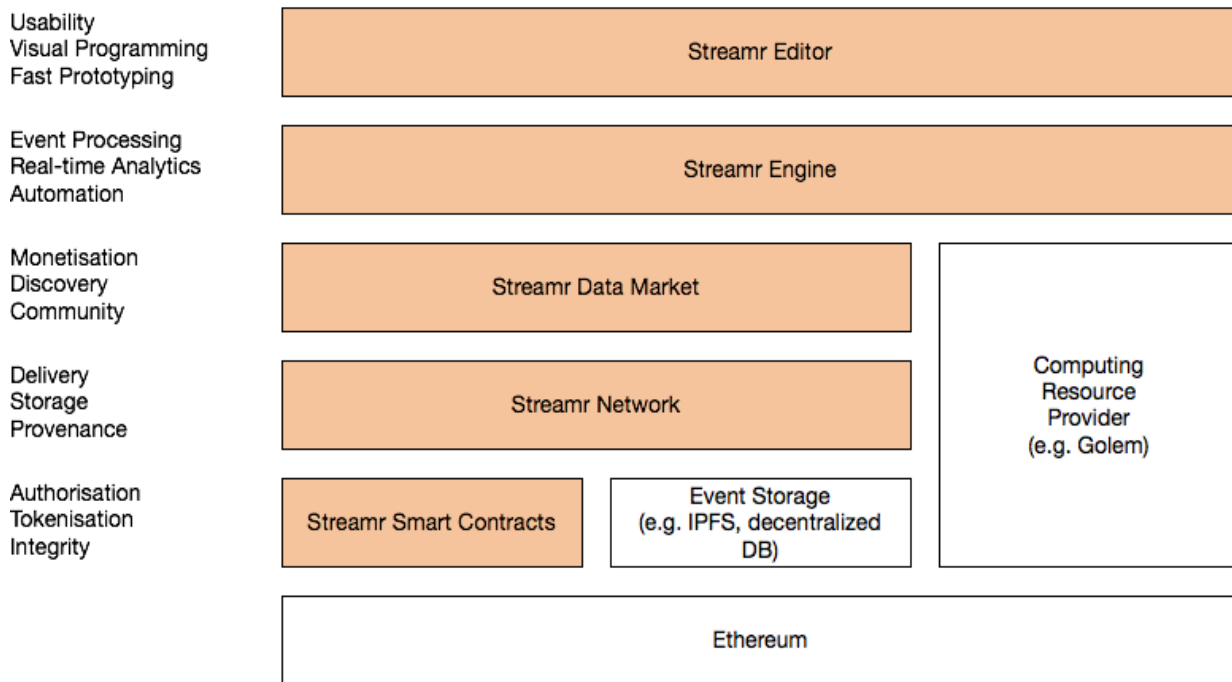


Abbildung 1. Streamr-Technologiebaum.

## 2.1 Streamr Editor

Der Streamr Editor ermöglicht die zügige Entwicklung von datengesteuerten intelligenten Verträgen, senkt die Kosten für die Erstellung von Dapps und enthält Vorlagen für häufige Anwendungsfälle.

In der Geschäftswelt herrscht momentan zwar ein großes Interesse an der Blockkette und dezentralen Anwendungen. Die Zahl der Fälle, in denen es um die Anwendung in der Praxis geht, ist jedoch nach wie vor sehr begrenzt. Wir befinden uns in den frühen Tagen einer neuen Technologie. Es ist daher nicht weit hergeholt zu behaupten, dass viele von denen, die sich in diesem Bereich engagieren wollen, keine Experten für die Feinheiten von Ethereum, des Prinzips der Solidität, Verschlüsselungsfragen, Datenprovenienz und anderen technischen Fragen sind.

Unseres Erachtens erfordert das kommerzielle Wachstum des Ökosystems Werkzeuge, die es auch Laien ermöglichen, intelligente Verträge zu erstellen, sich mit vertrauenswürdigen Datenquellen zu

verbinden, sichere, blockkettenunabhängige Module für die Datenfilterung, -aggregation und Verfeinerung zu nutzen, dezentrale Anwendungen einzusetzen, die Ausführung intelligenter Verträge zu verfolgen und den Fluss von Eingangsdaten und Blockketten-Events zu visualisieren.

Wir gehen auf die Notwendigkeit einer Usability-Schicht ein, indem wir leistungsstarke Werkzeuge (wie einen benutzerfreundlichen visuellen Editor), Wrapper und Vorlagen für intelligente Verträge anbieten, die sich an Domainprofis und geschäftliche Anwender richten. Diese Werkzeuge verbergen die Technologie tief unter ihrer Motorhaube, übernehmen die Datenintegration und -kommunikation und automatisieren die Routineschritte bei der Bereitstellung und Überwachung von intelligenten Verträgen.

Wir stellen uns ein Ökosystem vor, in dem es mehrere Usability-Plattformen und -Werkzeuge gibt. Die bestehende Streamr-Plattform implementiert bereits einige Elemente der Usability-Schicht, wobei in den kommenden Monaten und Jahren weitere Funktionen hinzukommen werden. Unser Ziel ist es, dass Sie in wenigen Minuten einen nützlichen und funktionierenden, datengetriebenen intelligenten Vertrag aufbauen und implementieren können. Das ist mehr als eine Phantasie; unsere Demo bei der EDCON<sup>5</sup> Paris im Februar 2017 ist ein Vorgeschmack auf das, was man alles tun kann. (siehe Abbildung 2 zur Veranschaulichung).

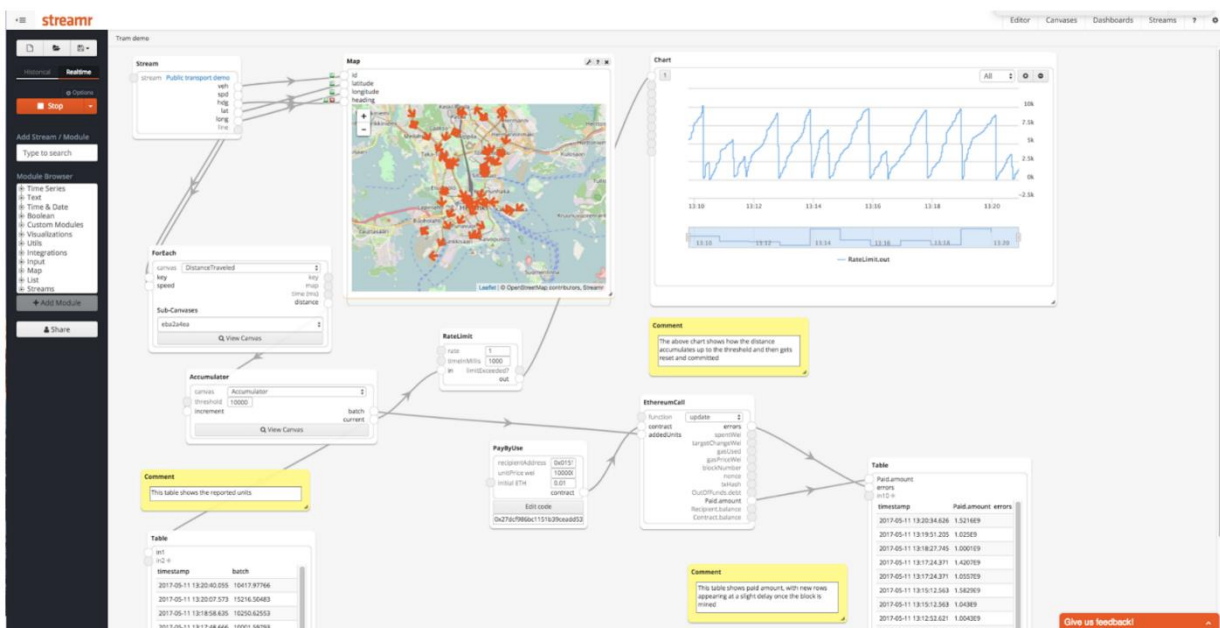


Abbildung 2. Eine Alpha-Version des Arbeitsbereichs des Streamr Editors.

Dies sind einige der geplanten Funktionen für die Usability-Schicht:

- Ein visueller Editor für die Erstellung intelligenter Verträge, die Einspeisung von Daten und den Bau von Pipelines für die Datenverarbeitung außerhalb der Blockkette.
- Module für die Kommunikation mit intelligenten Verträgen und die Interaktion mit der

<sup>5</sup>Henri Pihkala: "Connecting Ethereum with the real world: How to easy create data-driven smart contracts", European Ethereum Development Conference (EDCON), Paris, 17.-18. Februar 2017, (<https://www.youtube.com/watch?v=C110rcj-Fok>)

Blockkette.

- Module für die blockkettenunabhängige Verarbeitung: Datenfilterung, Datenveredelung und -aggregation, Einsatz dezentraler Anwendungen, Überwachung der Abwicklung intelligenter Verträge und Visualisierung der Inputströme und Events auf der Blockkette.
- Ein Solidity-Editor, in dem der Code für intelligente Verträge in einer kontextsensitiven Umgebung geschrieben und modifiziert werden kann.
- Eingebaute und getestete Open-Source Solidity-Vorlagen für verschiedene Anwendungsfälle von intelligenten Verträgen auf Ethereumbasis.
- Wiedergabefunktionalität zur Simulation der Funktionalität von intelligenten Verträgen, zum Debuggen des Vertragscodes und zum Testen der Funktionalität vor der Bereitstellung.
- Ein-Klick-Funktionalität, um einen intelligenten Vertrag entweder in einem Testnetz oder im Hauptnetz zu versenden.

## 2.2 Streamr Engine

Die Streamr Engine ist das leistungsstarke Analysewerkzeug, das blockkettenunabhängig innerhalb eines Anbieters für dezentrale Rechenleistung (z.B. in einem Docker-Container auf Golem) ausgeführt wird.

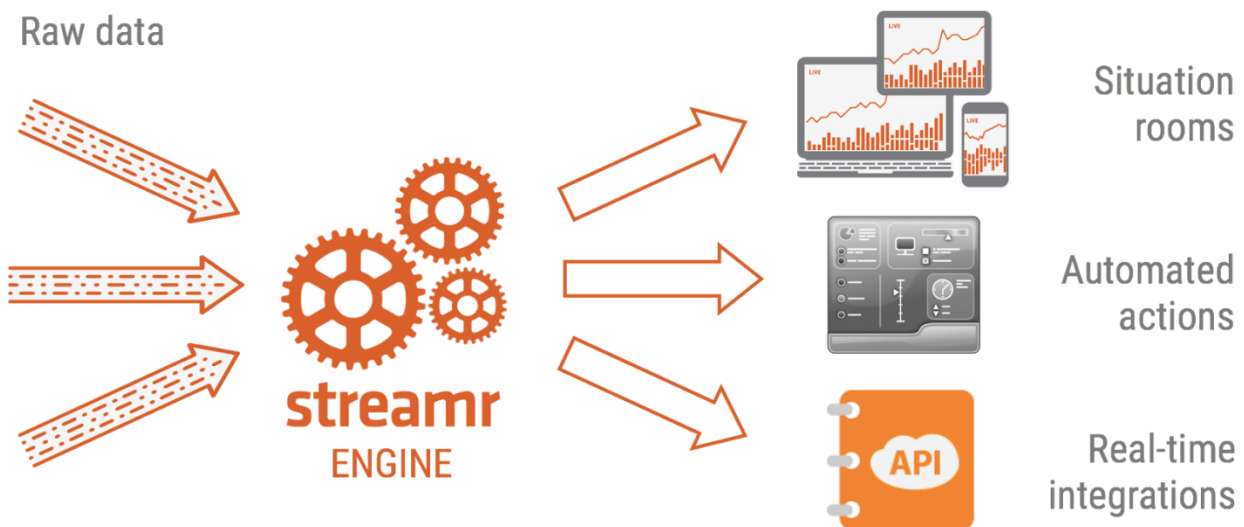


Abbildung 3. Typische Datenflussmuster und -ergebnisse für die Analyse-Engine von Streamr.

Dapps, meist mit webbasierten Benutzeroberflächen und Backends, welche auf intelligenten Verträgen basieren, haben derzeit keine Möglichkeit, Rohdaten zu verarbeiten und in Informationen zu verwandeln. Eine Gruppe von IoT-Sensoren oder eine Finanzbörse kann jedoch Tausende oder sogar Millionen von Ereignissen (Events) pro Sekunde produzieren; eine Menge, deren Berechnung eine Blockkette zweifellos überfordern oder die Kosten dafür ins Unermessliche steigen lassen würde.

Eine Streaming-Analyse-Ebene wird benötigt, um Rohdaten in Informationen zu verwandeln und für Dapps und intelligente Verträge nutzbar zu machen. Rohdaten müssen unter Umständen gefiltert, gesampled, aggregiert oder mit anderen Daten kombiniert werden, oder sie müssen Algorithmen zur Erkennung von Anomalien durchlaufen oder durch fortgeschrittenes maschinelles Lernen und

Mustererkennungsmodelle verarbeitet werden. Oder Sie möchten vielleicht Dinge tun, die mit intelligenten Verträgen einfach nicht möglich sind, z.B. externe APIs als Teil der Verarbeitungskette aufrufen.

Die Streamr Engine beobachtet Events im Streamr-Netzwerk. Modelle, die mit dem Streamr Editor erstellt wurden, verfeinern eingehende Daten und reagieren in Echtzeit auf neue Events. Es gibt viele Möglichkeiten, wie diese Reaktionen aussehen können, einschließlich der folgenden:

- Veröffentlichung von bereinigten oder verfeinerten Daten in einem anderen Stream im Streamr-Netzwerk, vielleicht in Echtzeit durch eine Dapp-Benutzeroberfläche dargestellt, die ebenfalls mit dem Netzwerk verbunden ist.
- Interaktion mit einem IoT-Gerät, z.B. zur Steuerung eines Auslösemechanismus, zum Öffnen eines Schlosses, zum Einschalten der Beleuchtung oder zum Rufen eines Aufzugs.
- Versenden eines Alarms per E-Mail oder Push-Benachrichtigung.
- Aufruf einer Funktion in einem intelligenten Vertrag.

Die Verwendung des Streamr-Netzwerks als Bindeglied für das Messaging zwischen Dapps und der blockkettenunabhängigen Rechenleistung innerhalb der Engine ermöglicht eine ganz neue Kategorie dezentraler Apps: Apps, die von nicht-trivialen Datenmengen angetrieben werden. Natürlich können die Ergebnisse auch von traditionellen, zentralisierten Anwendungen verarbeitet werden, während die Vorteile dezentraler Messaging- und Analysefunktionen erhalten bleiben.

### 2.3 Streamr Data Marketplace

Der Datenmarktplatz von Streamr (Streamr Data Market) ist ein globales Universum von gemeinsam genutzten Datenströmen, zu denen jeder beitragen und die jeder abonnieren kann. Es ist ein Ort der Datenmonetarisierung und des Datenaustausches auf der Machine-to-Machine (M2M)-Ebene. Der Datenmarktplatz bietet Anonymität, ermöglicht aber bei Bedarf die Überprüfung der digitalen Identität.

Der Datenmarktplatz ist ein Treffpunkt für Datenproduzenten und Datenkonsumenten. Hier können Konsumenten die angebotenen Daten verwenden und darauf zugreifen, um sie als Input für Dapps, intelligente Verträge oder traditionelle Apps zu nutzen.

Die Daten sind in Datenströmen organisiert, dem Grundbaustein des Datenmarktplatzes und einem Basiselement im Streamr-Netzwerk (siehe Kapitel 2.4 unten). Datenströme enthalten Events aus Datenquellen, die in regelmäßigen oder unregelmäßigen Abständen neue Datenpakete (Data Points) aussenden. Hier sind einige typische Situationen, bei denen Echtzeitdaten im Streamingverfahren erzeugt werden:

- Eine Börse erzeugt jedes Mal, wenn es ein neues Gebot oder Angebot gibt, und jedes Mal, wenn ein Handel stattfindet, ein neues Event.
- Ein Fahrzeug des öffentlichen Nahverkehrs übermittelt alle paar Sekunden seine Identität, seinen Status, seine Geschwindigkeit, seine Beschleunigungsdaten, seine Geolokalisierung und seinen Kurs.
- Ein Bewegungsmelder sendet ein Signal, wenn ein sich bewegendes Objekt in seiner Reichweite erkannt wird.



- IoT-Sensoren, die an einem elektrischen Antrieb angebracht sind, messen die Temperatur, Geschwindigkeit und Vibrationsstärke während des Betriebs in einer intelligent vernetzten Fabrik.
- Luftqualitätssensoren messen den Anteil von Kohlenmonoxid, Schwefeldioxid, Stickstoffdioxid und die Ozonwerte in städtischen Gebieten.
- Seismografen messen die Bewegungsstärke in einem Gebiet mit vulkanischer Aktivität.
- Intelligente Kleidung, die von Profisportlern getragen wird, sammelt biometrische Daten zum Herzschlag, der Körpertemperatur und der Beschleunigung.

Der Datenmarktplatz stellt eine große Auswahl an vertrauenswürdigen, mit Zeitstempeln versehenen Daten zum Abonnement zur Verfügung. Einige der Daten stammen von etablierten und professionellen Datenanbietern und Verteilern, andere von öffentlichen oder offen zugänglichen Datenquellen. Wichtig ist, dass die Plattform es jedem ermöglicht, seine Daten einzubringen und zu monetarisieren. Während Unternehmen wertvolle Daten von Sensoren und Geräten einfließen lassen, produzieren auch Privatpersonen wertvolle Informationen.

Zum Beispiel könnten Menschen, die eine Smartwatch tragen, ihre Herzfrequenzdaten auf dem Datenmarktplatz zum Verkauf anbieten. Die Daten können anonym angeboten werden, sodass die Privatsphäre nicht verletzt wird. Wer würde sich für solche Daten interessieren? Ein Unternehmen aus dem Pharmabereich könnte sie für Forschungszwecke kaufen, oder eine allgemeine Gesundheitspflegeorganisation könnte sie verwenden, um herauszufinden, wie häufig Menschen sportlich aktiv sind oder wie hoch die allgemeine Stressbelastung in der Bevölkerung ist. Ein Smartwatch-Hersteller könnte die Daten erwerben, um einen Eindruck zu gewinnen, wie seine Herzfrequenz-Sensoren arbeiten. Und die Datenproduzenten generieren allein dadurch ein tägliches Einkommen, dass sie ihre Daten zur Verfügung stellen.

Es gibt eigentlich keinen Grund dafür, warum Abonnements im Datenmarktplatz von Softwareentwicklern, Dateningenieuren oder Wissenschaftlern initiiert werden sollten. Tatsächlich könnte der dezentrale Markt letztendlich von Machine-to-Machine-Transaktionen dominiert werden. Autonome Maschinen, Roboter und intelligente Geräte benötigen Daten für ihren Betrieb und sie produzieren Daten, die für andere Teilnehmer im Ökosystem wertvoll sind.

So entstehen automatische, wertschöpfende Veredelungsmuster. Eine künstliche Intelligenz könnte zum Beispiel Rohmaterial für den Aktienmarkt abonnieren, eine proprietäre Mustererkennung anwenden um Handelssignale zu generieren und diese Signale auf demselben Datenmarktplatz zum Verkauf anbieten.

Während ein Großteil der Inhalte des Datenmarktplatzes für alle frei verfügbar sein wird, wird es auch Daten geben, die bezahlt werden müssen, und es wird Daten geben, für die eine Endbenutzerlizenz gilt. In solchen Fällen ist eine Abonnementlizenz erforderlich. Eine Lizenz berechtigt zum Zugriff auf die Daten für einen bestimmten Zeitraum - unter bestimmten Bedingungen und gegen eine Gebühr. Es gibt eine enge Analogie zum Streaming von Musik: man kann die abonnierten Daten nicht besitzen, genauso wenig wie man die Rechte an einem Song besitzt, indem man ihn auf Spotify hört oder ihn von iTunes herunterlädt.

Datenlizenzen werden als intelligente Verträge implementiert (siehe Abschnitt 2.5.4). Der große Vorteil der Blockkette besteht darin, dass sie eine vertrauenswürdige und dezentrale Möglichkeit bietet, die Nutzungsbedingungen und Zugriffsrechte zu speichern und dabei sicher zu stellen, dass die Datenzahlungen vereinbarungsgemäß erfolgen.

In einem breiteren Kontext enthält der Markt auch ein großes Potenzial für einen starken Netzwerkeffekt. Je mehr Inhalte vorhanden sind, desto attraktiver wird das Angebot sowohl für Datenlieferanten als auch für Datenkonsumenten. Im Datenmarkt von Streamr erleichtert ein Webportal (implementiert als Dapp) das Auffinden der dort vorhandenen Daten, stellt eine umfassende Werkzeugsammlung für die Erstellung und Verwaltung von Datenströmen zur Verfügung und erleichtert das Abonnieren von Datenströmen Ihrer Wahl.

## 2.4 Streamr Network

Das Streamr-Netzwerk ist die Datentransportschicht im Technologiebaum. Es besteht aus Streamr-Broker-Knoten, die ein P2P-Netzwerk aufbauen. Das Netzwerk verfügt über einen Publish/Subscribe-Mechanismus und unterstützt die dezentrale Speicherung von Events. Der Netzwerkdurchsatz skaliert linear mit der Anzahl der teilnehmenden Knoten und kann Millionen von Ereignissen pro Sekunde verarbeiten.

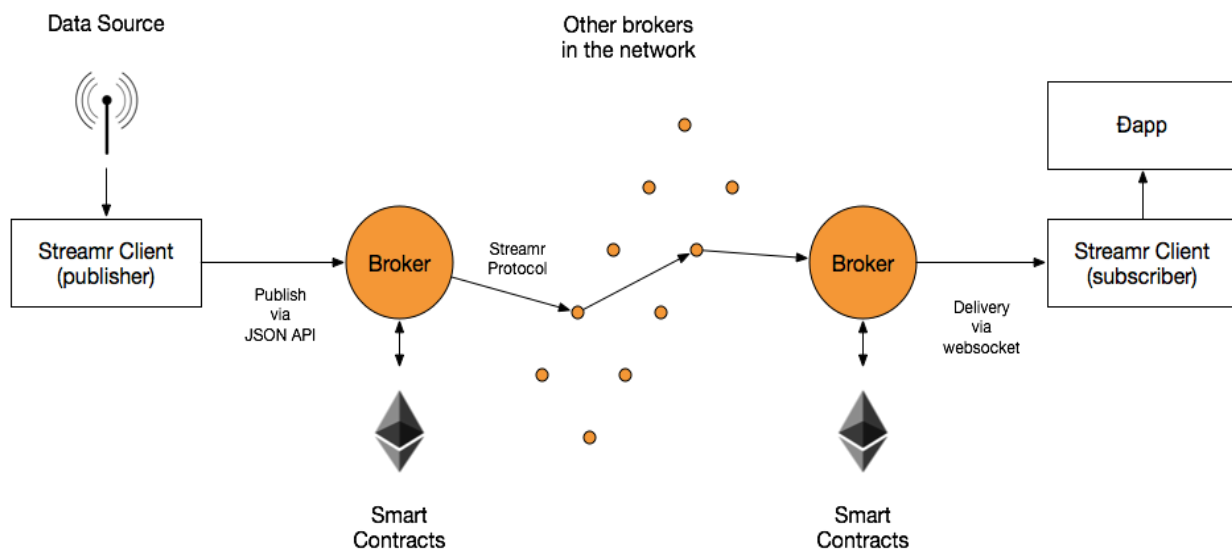


Abbildung 4. Ein Beispiel für ein Event, das durch das Brokernetzwerk von einer Datenquelle zu einer Subscriber-Dapp transferiert wird.

Das Streamr-Netzwerk (Abbildung 4) ist eine Transportebene des Streamr-Technologiebaums. Das Netzwerk übernimmt das gesamte Messaging in der dezentralen Datenpipeline. Die Ebene besteht aus Primitiven (Events und Streams) und Brokerknoten. Die Brokerknoten operieren auf der Basis der Primitive, und die Sammlung von Brokerknoten bildet das P2P-Netzwerk, welches die dezentrale Speicherung und das dezentrale Messaging verwaltet.

Die Infrastrukturebene verwendet die Ethereumtechnologie für ihren Betrieb. Die Knotenkoordination erfordert einen soliden Konsens, der durch intelligente Verträge umgesetzt wird. Die rohen Eventdaten wandern in der Regel nicht auf die Blockkette, was - zusammen mit der Partitionierung - dem Streamr-Netzwerk erlaubt, Millionen von Events pro Sekunde und mehr zu skalieren.

Das Streamr-Netzwerk vereint die besten Teile skalierbarer, cloud-basierter Echtzeit-Datentransporte (z.B. Kafka, ZeroMQ, ActiveMQ) mit dem, was in der dezentralen P2P/Crypto-Community (Whisper<sup>6</sup>, Bitmessage<sup>7</sup>) verfügbar ist. Die cloud-basierten Frameworks nutzen effizientes Sharding und Persistenzschemata, um einen hohen Durchsatz zu erreichen; sie tun dies jedoch nur in einer vertrauenswürdigen, lokalen Netzwerkumgebung.

Die Peer-to-Peer-Protokolle zeigen effektive Strategien für das Routing, das Finden neuer Peers, NAT-Traversal, Standortverschleierung usw. auf, liefern aber nicht den Durchsatz, der für datenintensive reale Anwendungen benötigt wird.

### 2.4.1 Events (Ereignisse)

Ein Event<sup>8</sup> ist eine mit einem Zeitstempel versehene Information. Jedes Event enthält Überschriften (Headers) und Inhalte (Content).

Die Header geben die Metadaten des Events an, wie z.B. Zeitstempel, Herkunft und Inhaltstyp. Das Event-Protokoll unterstützt beliebige Typen und Formate, z.B. JSON-Nachrichten oder Binärbilder. Der Inhaltstyp gibt an, in welchem Format der Inhalt vorliegt. Event-Header und Inhalte werden für die Übertragung in einem Binärformat kodiert.

Alle Events im Streamr-Netzwerk sind kryptographisch signiert. Alle Events haben zudem einen Ursprung, z.B. eine Ethereumadresse. Eine Signatur wird aus einem privaten Schlüssel und dem Rest der Nachricht berechnet. Die Signatur dient zum Nachweis der Herkunft der Nachricht sowie der Integrität der Nachricht. Da das Event-Format jede Art von Herkunft und Signatur zulässt, ist das System auch für die Zukunft sicher.

Die folgende Tabelle listet die Informationen auf, die in einem Event enthalten sind.

Feld	Beschreibung
version	Version des Event-Protokolls
stream	Stream-ID (Ethereumadresse des mit einem Stream verbundenen intelligenten Vertrags)
partition	Stream-Partition (siehe Abschnitt über Partitionierung)
timestamp	Event-Zeitstempel (ISO 8601)
contentType	Anleitung zum Parsen des Bodys (z.B. JSON)
encryptionType	Verschlüsselungsalgorithmus, der zur Verschlüsselung des Inhalts verwendet wird
content	Datennutzlast
originType	Anweisung, wie der Ursprung zu interpretieren ist
origin	Ursprung bzw. Urheber der Daten
signatureType	Anleitung zur Interpretation der Signatur
signature	die kryptographische Signatur beweist die Herkunft und Integrität der Nachricht

<sup>6</sup> Gav Wood: "WhisperPoC 2 Protocol Spec" (<https://github.com/ethereum/wiki/wiki/Whisper-PoC-2-Protocol-Spec>)

<sup>7</sup> Siehe [https://bitmessage.org/wiki/Main\\_Page](https://bitmessage.org/wiki/Main_Page).

<sup>8</sup> Streamr-Events sollten nicht mit Events in intelligenten Verträgen auf Ethereumbasis verwechselt werden.

## 2.4.2 Streams

Alle Events gehören zu einem Stream. Es kann beliebig viele Streams geben, die jeweils solche Events zusammenfassen, welche logisch zusammenhängen und in aufsteigender chronologischer Reihenfolge gespeichert sind.

Stream-Metadaten werden in einem intelligenten Vertrag auf Ethereumbasis gespeichert. Jeder Stream wird durch die Ethereumadresse des entsprechenden Vertrags identifiziert. Aus Gründen der Skalierbarkeit werden Events (d.h. die eigentlichen Datenpunkte) nicht in intelligenten Verträgen oder auf der Blockkette gespeichert.

Ein Datenstrom enthält eine Reihe von Berechtigungen. Die Berechtigungen steuern, wer Events aus dem Stream abonnieren kann (Subscribe) und wer neue Events im Stream veröffentlichen kann (Publish). Die Stream-Eigentümer kontrollieren die Berechtigungen, aber sie können die Berechtigungskontrolle auch an Dritte vergeben oder delegieren, wenn nötig.

Die folgende Tabelle listet die Metadaten für einen Stream auf.

Feld	Beschreibung
id	Stream-ID (Ethereumadresse)
name	Name des Streams
description	Beschreibung des Streams
owner	Eigentümer des Streams
permissions	Eine Abbildung der Struktur von der Ethereumadresse bis zu den Berechtigungsebenen

## 2.4.3 Veröffentlichen/Abonnieren (Publish/ Subscribe)

Der Datentransport im Netzwerk folgt dem Publish/Subscribe-Muster<sup>9</sup>. Events, die in einem Stream veröffentlicht werden, werden umgehend an alle autorisierten und verbundenen Abonnenten des Streams weitergeleitet. Das Abonnieren von Streams kann auf bestimmte Benutzer beschränkt werden oder für die Öffentlichkeit kostenlos freigegeben werden. Ebenso kann die Berechtigung, Inhalte in einem Stream zu veröffentlichen, von einer oder mehreren Personen oder gar von jedem gehalten werden.

Das Publish/Subscribe-Muster ermöglicht viele Messaging-Topologien, die in realen Anwendungen zum Einsatz kommen:

- One-to-many (z.B. Nachrichtensender oder Börsenticker)
- Many-to-many (z.B. ein Gruppenchat oder ein Multiplayer-Spiel)
- One-to-One (z.B. ein privater Chat oder eine Analyse-Pipeline)
- Many-to-one (z.B. ein Abstimmungssystem)

Beachten Sie, dass die Veröffentlichung eines Events nicht bedeuten muss, dass es auch an einen beliebigen Kunden ausgeliefert wird: es kann sein, dass es keine Abonnenten gibt. Dennoch wird das Event persistiert und zu Redundanzzwecken an eine Reihe von Brokerknoten weiter gegeben.

---

<sup>9</sup>Wikipedia: Publish/subscribe pattern ([https://en.wikipedia.org/wiki/Publish-subscribe\\_pattern](https://en.wikipedia.org/wiki/Publish-subscribe_pattern))

Von einem technischen Standpunkt aus gesehen gibt es zwei Arten von Abonnenten. Der größte Teil des Datenflusses geht an Teilnehmer, die über eine direkte Verbindung zu einem Brokerknoten an das Netz angeschlossen sind (siehe Abschnitt 2.4.4. unten). Dies können z.B. Web-Frontends von Dapps, Eventverarbeitungsketten, die auf der Streamr-Engine laufen, oder IoT-Geräte sein, die von Daten aus dem Netzwerk gesteuert werden.

Intelligente Verträge sind eine besondere Art von Abonnenten, die das Streamr-Netzwerk ermöglicht. Broker-Knoten im Netzwerk werden dazu angeregt, Events an intelligente Verträge, die als Abonnenten auftreten, auszuliefern. In diesem Szenario gelten natürlich die Skalierbarkeitsgrenzen von Blockketten. Der Mechanismus ermöglicht es dem Netzwerk, als „Orakel“ zu agieren. Das bedeutet, dass Daten ohne Hilfe eines Drittanbieters in intelligente Verträge verschoben werden können. Da alle Daten im Netzwerk an der Quelle signiert werden, können sie jederzeit verifiziert und als vertrauenswürdig angesehen werden.

#### **2.4.4 Brokerknoten**

Der Streamr-Brokerknoten ist die zentrale Softwarekomponente im Netzwerk. Ein Brokerknoten übernimmt Aufgaben wie das Veröffentlichen von Events, das Abonnieren von Streams, das Verwalten von Speicherplatz und die Kommunikation mit Ethereumknoten über JSON RPC-Aufrufe. Der Brokerknoten stellt den angeschlossenen Anwendungen seine Funktionalität über APIs zur Verfügung.

Die Broker-API kann von Anwendungen bzw. Apps aus verwendet werden, die Standard-HTTP- und WebSocket-Bibliotheken in einer beliebigen Sprache verwenden. Um die Bedienung zu vereinfachen, stellen wir zudem Referenzimplementierungen in verschiedenen Sprachen zur Verfügung.

Die primäre Client-Bibliotheksplattform wird in JavaScript geschrieben. Sie kann verwendet werden, um Daten an webbasierte Dapps zu liefern, die im Browser laufen, sowie an Back-End-Anwendungen, die node.js ausführen. Eine WebSocket-API führt die Eventübermittlung von Datenquellen an das Netzwerk und vom Netzwerk an Dapps aus. Für die Verwaltung von Streams wird eine JSON-API verwendet.

Die WebSocket-Streaming-API übernimmt die folgenden Aufgaben:

- Authentifizierung einer Sitzung
- Veröffentlichung von Events
- Abonnement von Events in Streams
- Übertragung von Events an abonnierende Kunden
- Suche nach historischen Events in Streams

Die JSON-API stellt die folgenden Funktionen zur Verfügung:

- Erstellen eines Streams
- Konfigurieren eines Streams
- Löschen eines Streams
- Finden von Informationen über einen Stream
- Streams nach Suchkriterien durchstöbern
- Veröffentlichen von Events (alternativ zur WebSocket-API)
- Suche nach historischen Events in Streams (alternativ zur WebSocket-API)

Der größte Teil des Datenverkehrs zwischen Brokern besteht aus Eventmeldungen, aber es entsteht auch Datenverkehr im Zusammenhang mit dem Routing und dem Finden neuer Peers. Eine wichtige Koordinationsaufgabe zwischen Brokern ist die Partitionierung, bei der ein verlässlicher Konsens erzielt werden muss. Dieser Mechanismus ist als intelligenter Vertrag implementiert, der die Kraft des Ethereum-Netzwerks nutzt (siehe Abschnitt 2.4.5).

#### **2.4.5 Partitionierung (Sharding)**

Der Eventverkehr im gesamten Netzwerk wird in unabhängige Teile, sogenannte Partitionen, aufgeteilt. Jeder Brokerknoten verwaltet den Verkehr, der zu einer Reihe von Partitionen gehört. Auf diese Weise wird die Skalierbarkeit erreicht: nicht alle Knoten bewältigen den gesamten Datenverkehr. Dies ist vergleichbar mit dem Partitionierungsschema, das z.B. in Apache Kafka verwendet wird.

Die Partition für ein bestimmtes Event wird durch das Hashing der Stream-ID berechnet. Dies ist ein schneller Vorgang und er wird lokal ausgeführt. Die Verwendung der Stream-ID als Partitionsschlüssel bedeutet, dass sich alle Events in einem bestimmten Stream immer innerhalb der selben Partition befinden. Dies ermöglicht es dem Netzwerk, die Reihenfolge der Events innerhalb eines Streams zu verwalten und diese effizient zu speichern.

Es kann vorkommen, dass ein Stream eine solche Menge an Nachrichten empfängt, dass ein einzelner Broker sie nicht verarbeiten kann. In diesem Fall wird eine weitere Partitionierungsrunde auf die Streams selbst angewendet, und der Verkehr innerhalb eines Streams wird in voneinander unabhängige Teile aufgeteilt. Anschließend werden die (Stream-ID, Stream-Partition) Tupel gehasht, um die Netzwerkpartition zuzuweisen, und der Publisher stellt den Partitionsschlüssel zur Verfügung, der das Event einer Partition innerhalb des Streams zuordnet. Die Reihenfolge der Events für einen Stream-Partitionsschlüssel bleibt erhalten.

Die Anzahl der Partitionen im Netzwerk bleibt konstant, bis sie im Laufe der Zeit automatisch erhöht wird. Wie im nächsten Abschnitt beschrieben, gibt es einen intelligenten Vertrag, der als Koordinator fungiert und die Netzwerkpartitionierung steuert. Die Anzahl der Partitionen ist proportional zur Anzahl der Brokerknoten, die am Netzwerk teilnehmen.

#### **2.4.6 Knotenkoordination**

In verteilten Datensystemen wie Apache Kafka und Apache Cassandra wird die Knotenkoordination in der Regel durch den Einsatz einer Komponente wie Apache Zookeeper erreicht. Es gibt einen zentralisierten Prozess für die Konsensfindung bei Vorgängen, wie zum Beispiel bei der Wahl von Führungskräften. Alternativ dazu verwenden einige Systeme die manuelle Zuweisung von Koordinatorknoten, die über besondere Privilegien im Netzwerk verfügen.

In einem dezentralen Netzwerk können solche zentralen oder privilegierten Komponenten nicht existieren. Stattdessen nutzt das Streamr-Netzwerk das Ethereum-Netzwerk, um einen Konsens für die Knotenkoordination im P2P-Netzwerk herzustellen.

Die zentrale Koordinierungsaufgabe ist die Zuordnung von Netzwerkpartitionen zu Brokerknoten im Netzwerk und die ordnungsgemäße Erfassung von Änderungen dieser Informationen, wenn Knoten erscheinen oder verschwinden. Anstatt dies durch eine zentrale Komponente wie Zookeeper zu

erreichen, wird diese Aufgabe durch einen intelligenten Vertrag umgesetzt: den Netzwerkkoordinator. Der intelligente Vertrag zur Netzwerkkoordination wird auf der Ethereum-Blockkette eingesetzt. Brokerknoten ermitteln den aktuellen Netzwerkstatus, indem sie den intelligenten Vertrag beobachten und abfragen. Die Aufrüstung des Netzes erfolgt einfach durch den Wechsel zu einem neuen Netzwerkkoordinatorvertrag.

Das Ausgleichen (Rebalancing) der Partitionszuweisungen ist eine der Aufgaben des Netzwerkkoordinatorvertrags. Es werden nur sinnvolle Änderungen vorgenommen, und wenn es keine gibt, führt die Funktion nichts aus. Wenn das Netzwerk asymmetrisch ist, überträgt der Aufruf der Funktion DATAcoin an den Urheber des Aufrufs. Dieser Anreizmechanismus stellt sicher, dass bei Bedarf eine Umstrukturierung des Netzwerks stattfindet.

Die Knoten, die einer Partition zugeordnet sind, erhalten alle Daten dieser Partition. Einige oder auch alle Knoten berechnen rollierende Prüfsummen auf der Grundlage der Daten und melden die Prüfsummen in bestimmten Zeitabständen an den intelligenten Vertrag in der Form des Netzwerkkoordinators. In einem großen öffentlichen Netzwerk gibt es für jede Partition genügend Knoten, um eine Kollision zu erschweren. Auch die Partitionsvergabe durch den Netzwerkkoordinator ist nur schwer zu beeinflussen.

#### 2.4.7 Anreizmechanismus

Abonnenten (Subscriber) sind die Konsumenten von Daten im Netzwerk. Mit DATAcoin, dem offiziellen Token des Netzwerks, können Abonnenten Streams beziehen. Andere Parteien gewinnen DATAcoin, indem sie zum Netzwerk beitragen: die Brokerknoten (die "Miner" dieses Netzwerks) und die Datenproduzenten.

Brokerknoten erhalten einen Anreiz dazu, zwei Dinge zu tun: Prüfsummen für ihre zugewiesenen Partitionen an den Netzwerkkoordinator (siehe Abschnitt 2.4.6. oben) zu melden und Daten an alle Abonnenten eines intelligenten Vertrags zu liefern (siehe Abschnitt 2.4.3). Beide Vorgänge kosten etwas Ethereum-Gas, das vom Broker bezahlt wird. Diese Kosten werden durch DATAcoin gedeckt, die die Makler für die Funktionsfähigkeit des Netzwerks erhalten

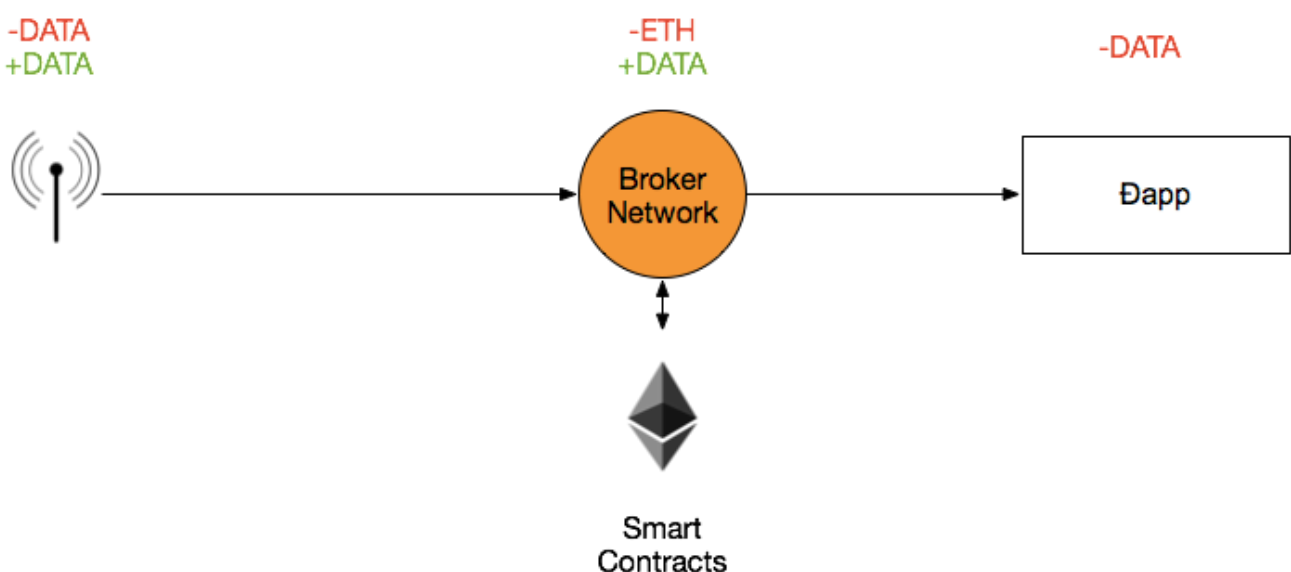


Abbildung 5. Schematische Darstellung der Anreizstruktur im Streamr-Netzwerk.

Prüfsummen für eine Partition werden von mehreren Brokerknoten berechnet und gemeldet, und die Broker werden nur dann belohnt, wenn sie sich auf die Prüfsummen über eine festgelegte Kohärenzschwelle einigen, die vorher im intelligenten Vertrag des Netzwerkkoordinators festgelegt wurde (z.B. müssen 90% der zugewiesenen Broker eine bestimmte Prüfsumme melden, damit sie als gültig angesehen werden kann). Wenn ein Knoten abweichende oder gar keine Prüfsummen meldet, oder wenn die Prüfsummen nicht kohärent sind, wird keine Belohnung ausbezahlt und die Wahrscheinlichkeit sinkt, dass diesem Knoten in Zukunft die Verantwortung für eine Partition zugewiesen wird.

Wie bereits erwähnt können auch intelligente Verträge Abonnenten eines Streams sein. Der Abonnent legt eine Prämie für die Zustellung von Events an den intelligenten Vertrag in der Form von DATAcoins fest. Die Prämie wird von demjenigen eingezogen, der die Daten zuerst liefert. In der Regel ist dies der Brokerknoten, der direkt mit dem Publisher verbunden ist, da dieser Broker in einer Vorreiterposition ist, um die Lieferung durchzuführen. Andere Knotenpunkte oder externe Abonnenten können diesen Prozess beobachten und Möglichkeiten zur Bereitstellung der Daten identifizieren, sofern sie nicht vom ursprünglich vorgesehenen Übermittler geliefert werden.

Außerdem ist ein Mechanismus erforderlich, um Überlastungen (Flooding) im Netzwerk zu verhindern. Ein minimaler Kostenaufwand muss mit allen Veröffentlichungsvorgängen und Lieferungen an Abonnenten verbunden sein. Das Netzwerk kann die Kosten sammeln und hin und wieder auf die zugrunde liegende Blockkette – zu Zwecken der Skalierbarkeit - zurückgreifen, ähnlich wie staatliche Kanäle oder Micropayment-Kanäle in einigen Blockketten-Netzwerken funktionieren.

#### **2.4.8 Eventpersistenz**

Events in Datenströmen werden im Peer-to-Peer-Netzwerk persistiert. Damit wird das Netzwerk zu einer dezentralen Zeitreihendatenbank. Die Dezentralisierung bringt eine Reihe von Vorteilen mit sich, darunter eine größere Robustheit, eine geringere Fehlertoleranz, geringere Kosten und die Möglichkeit zur Anonymisierung.

Da es sich bei Streams um Sequenzen von Events handelt, ist die einfachste Form der Speicherung ein Eventprotokoll. Ein solches Eventprotokoll kann in einem beliebigen Blockspeicher abgelegt werden, z.B. in einem Dateisystem auf den Knoten selbst oder in einem dezentralen Objektspeicher wie Swarm<sup>10</sup>, IPFS<sup>11</sup> oder Storj<sup>12</sup>.

Für eine wesentlich größere Detailgenauigkeit und auch für bessere Abfragemöglichkeiten entstehen dezentrale Datenbanken wie BigchainDB<sup>13</sup>. Eine Lösung wie diese ist ein wahrscheinlicher Kandidat für die Eventspeicherung im Stream-Netzwerk. Die Rahmenbedingungen verändern sich jedoch rapide, und wir werden uns zum jetzigen Zeitpunkt nicht auf eine bestimmte Speicherlösung festlegen.

#### **2.4.9 Datenprovenienz**

Die Sicherheit und Herkunft von Daten ist immer dann von entscheidender Bedeutung, wenn externe Daten als Input für Dapps und intelligente Verträge verwendet werden. Da Transaktionen auf der

---

<sup>10</sup>Viktor Trón et al: "Introduction to Swarm" (<http://swarm-guide.readthedocs.io/en/latest/introduction.html>)

<sup>11</sup>Siehe "IPFS - The Permanent Web" (<https://github.com/ipfs/ipfs>)

<sup>12</sup>Siehe "Storj: A Peer-to-Peer Cloud Storage Network" (<https://storj.io/storj.pdf>).

<sup>13</sup>Siehe "BigchainDB: A Scalable Blockchain Database" (<https://www.bigchaindb.com/whitepaper/>)



Blockkette unwiderruflich sind, gibt es einen klaren Anreiz für ehrliche Parteien dafür zu sorgen, dass der Input vertrauenswürdig ist. Es gibt auch einen Anreiz für unlautere Parteien - ebenso wie für skrupellose Hacker - die Daten zu Geldzwecken zu manipulieren.

Im Streamr-Netzwerk wird jeder Datenpunkt kryptographisch durch einen privaten Schlüssel signiert, der sich im Besitz der Urheber der Daten befindet. Dieser Bereich befindet sich in einer rasanten Entwicklung und es sind viele verschiedene Signaturmethoden möglich. Events können z.B. mit einem privaten Ethereumschlüssel, einem X.509-Zertifikat, das einem IoT-Sensor gehört, durch vertrauenswürdige Hardware-Enklaven, die den Intel SGX-Chip und das Town Crier<sup>14</sup>-Relais verwenden oder durch einen TLSNotary<sup>15</sup>-Dienst, der Daten von einer Web-API überbrückt, signiert werden.

Unvoreingenommenheit und Flexibilität gegenüber der Methode, mit der die Datenprovenienz nachgewiesen wird, ist eines der grundlegenden Merkmale des Streamr-Netzwerks. Es kann in der Tat jede Methode unterstützen, die heute oder in der Zukunft verfügbar ist. Events im Netzwerk tragen immer sowohl die Signatur selbst als auch Informationen darüber, welche Methode verwendet werden soll, um die Signatur zu verifizieren. Die Client-Bibliotheken, die Events veröffentlichen und abonnieren, können schrittweise eine Unterstützung für verschiedene Methoden hinzufügen, indem sie die Struktur jeder neuen Methode abstrahieren und die Signaturprüfung aus der Sicht der Entwickler vereinfachen.

Der anfänglich unterstützte Signaturalgorithmus ist derselbe secp256k1 ECDSA, der auch von Ethereum verwendet wird. Auf diese Weise kann das Netzwerk alle veröffentlichten Daten zuverlässig auf einer Ethereumadresse abbilden.

#### 2.4.10 Vertraulichkeit der Daten

Da jeder am Streamr-Netzwerk teilnehmen kann, indem man einen Brokerknoten betreibt, werden Event-Datenströme von nicht-öffentlichen Streams im Streamr-Netzwerk mit asymmetrischer Schlüsselkryptographie stark verschlüsselt. Nur wer einen autorisierten privaten Schlüssel besitzt, kann die Daten lesen. Ein intelligenter Stream-Vertrag enthält den öffentlichen Schlüssel derjenigen, die auf den Stream zugreifen dürfen. Zum Zeitpunkt der Veröffentlichung werden die öffentlichen Schlüssel der berechtigten Empfänger verwendet, um die Daten zu verschlüsseln, sodass nur autorisierte Empfänger auf die Daten zugreifen können.

Multicast-Verschlüsselungstechniken<sup>16, 17</sup> können verwendet werden, um die Größe der Nachricht mit der Komplexität der Verschlüsselung in Einklang zu bringen. Die integrierte Verschlüsselungsunterstützung ermöglicht zudem eine unkomplizierte Datenmonetarisierung, da Dienste wie der Streamr Datenmarktplatz geschaffen werden können, um den Zugang zu Streams zu

---

<sup>14</sup>Fan Zhang et al.: "Town Crier: An Authenticated Data Feed for Smart Contracts", Berichte von der 2016er ACM SIGSAC Konferenz „Computer and Communications Security (CCS)“, Wien, Österreich, 24.-28. Oktober 2016, Seiten 270-282 (<https://eprint.iacr.org/2016/168.pdf>)

<sup>15</sup>“TLSnotary - a mechanism for independently audited https sessions”, Whitepaper, 10. September 2014 (<https://tlsnotary.org/TLSNotary.pdf>)

<sup>16</sup> Micciancio, Daniele und Saurabh Panjwani. „Multicast Encryption: How to maintain secrecy in large, dynamic groups?“ (<http://cseweb.ucsd.edu/~spanjwan/multicast.html>)

<sup>17</sup> Duan, Yitao und Canny, John. Computer Science Division, Universität von Berkeley. „How to Construct Multicast Cryptosystems Provably Secure Against Adaptive Chosen Ciphertext Attack“. (<http://www.cs.berkeley.edu/~jfc/papers/06/CT-RSA06.pdf>)

verkaufen oder zu vermieten. Publisher können den Datenstrom je nach Bedarf mit neuen Schlüsseln versehen, um den Zugriff selektiv zu sperren, z.B. wenn sie Abonnenten erwischen, die ihre Daten außerhalb des Netzwerks weiterverkaufen.

Ein dezentraler Ansatz in Kombination mit der Verschlüsselung fördert die Sicherheit. Die Daten werden auf eine Vielzahl von unbekanntem physischen Orten fragmentiert verteilt und durch eine starke Verschlüsselung während des Transports und der Speicherung geschützt. Dieser Ansatz adressiert die Ängste von Unternehmen und Organisationen, die sich Sorgen über das Potenzial für kompromittierte Daten durch den physischen Zugang zu Rechenzentren und Speicheranlagen machen.

## 2.5 Streamr - Intelligente Verträge

Eine Reihe von intelligenten Verträgen auf Ethereumbasis hilft beim Betrieb des Streamr-Netzwerks und des Datenmarktplatzes. Das Streamr-Netzwerk nutzt intelligente Verträge zur Anreizbildung, Koordination, Authorisierung und Integritätsprüfung. Der Datenmarktplatz baut auf den Funktionen auf, die das Netzwerk für die Lizenzierung und Monetarisierung von Daten zur Verfügung stellt. DATAcoin, ein ERC20-Token, wird von beiden Ebenen zu Anreizzwecken, als Reputationsmetrik und als Zahlungsmittel verwendet.

### 2.5.1 Stream

Der **intelligente Vertrag für einen Stream** enthält Informationen über diesen (siehe Abschnitt 2.4.2). Neben statischen Informationen enthält er auch die Berechtigungen für den Stream. Insbesondere trägt er die öffentlichen Schlüssel von berechtigten Empfängern für verschlüsselte Streams, die möglicherweise an eine Datenlizenz gebunden sind (siehe unten).

### 2.5.2 Stream-Registrierung

Der **Stream-Registry-Vertrag** enthält Informationen über bekannte Streams im Netzwerk. Streams können der Registry für Suchzwecke hinzugefügt werden. Die Stream-Registry kann auch Streams auf Basis des ENS (Ethereum Name Service) registrieren.

### 2.5.3 Netzwerkkoordinator

Der **Netzwerkkoordinator - in der Form eines intelligenten Vertrags** - weist den Brokerknoten Partitionen zu (siehe Abschnitt 2.4.6).

Brokerknoten melden sich beim Koordinator an und erhalten durch die Überwachung des intelligenten Vertrages aktuelle Informationen über den Netzwerkstatus.

### 2.5.4 Datenlizenz

Der intelligente **Vertrag zur Datenlizenz** stellt ein Produkt dar, das im Streamr Datenmarktplatz aufgeführt ist. Als Gegenleistung für DATAcoins gewährt der Vertrag den Zugang zu einer Reihe von Streams, indem der öffentliche Schlüssel des Käufers bei den Streams registriert wird. Die Datenlizenz kann für einen bestimmten Zeitraum gültig sein. Nachdem die Lizenz abgelaufen ist, hat der Käufer keinen Zugriff mehr auf neue Daten, die auf dem/den

Stream(s) veröffentlicht wurden.

Die Existenzberechtigung eines Lizenzvertrages besteht in der Erbringung des Nachweises, dass der Empfänger das Recht hat, zu bestimmten und unveränderlichen Nutzungsbedingungen auf einen Stream zuzugreifen. Gleichzeitig soll die Lizenz gewährleisten, dass der Datenlieferant die vereinbarte Vergütung für Echtzeitdaten erhält, sobald er diese veröffentlicht. Die Nutzungsbedingungen können im Datenlizenzvertrag entweder direkt (und nach Bedarf gehasht) oder als Link zu einer inhaltsadressierten Speichermöglichkeit wie IPFS gespeichert werden. Der Vertrag kann auch Nachweise über erfüllte gesetzliche Anforderungen enthalten, wie z.B. das Ergebnis eines Know-Your-Customer (KYC)-Prozesses.

Brokerknoten melden Eventzahlen und rollende Hashes an intelligente Verträge, welche diese wiederum an die zugehörigen Lizenzverträge melden können. Die Lizenzverträge können nahezu beliebige Schutzvorkehrungen zur Verhinderung von Betrug vorsehen. Sie können z.B. die Zahlung sperren, bis eine bestimmte Anzahl von Events im Stream veröffentlicht wurde. Die Zahlung könnte auch schrittweise im Laufe der Zeit oder in einem Event-by-Event Ablauf erfolgen, wenn diese Events veröffentlicht werden. Vielleicht wird es auch einen Mechanismus für Abonnenten geben, um schlechte Daten zu kennzeichnen, was sich negativ auf die Reputation des Publishers auswirkt (siehe Abschnitt 3 zu DATAcoin und Karma). Diese Merkmale stellen sicher, dass der Publisher nicht bezahlt werden kann, ohne die versprochene Qualität der Daten zu liefern.

### 3. DATAcoin

DATAcoin ist das Mittel der Kompensation zwischen Datenproduzenten und Konsumenten. Das Token verkörpert auch den Anreiz für den Betrieb von Brokerknoten im P2P-Netzwerk. DATAcoin ist die Grundlage für Karma, die Reputationsmetrik in der Community. In einem größeren Zusammenhang ist es eine Möglichkeit, Daten als wertvolles Gut zu nutzen.

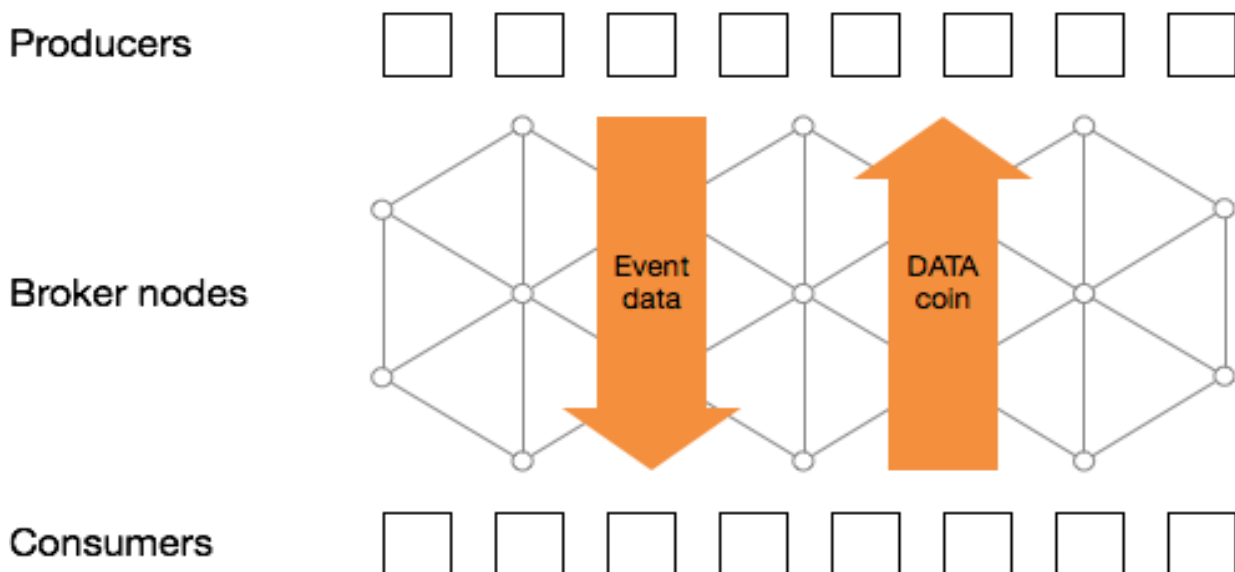


Abbildung 6. DATAcoin fließt in die entgegengesetzte Richtung der Daten.

In der dezentralen Datenpipeline nimmt das digitale Token eine wichtige Rolle ein. DATAcoin ist das Verbrauchstoken des Streamr-Netzwerks. DATA ist das Symbol des Tokens.

- Die Wartung und der Betrieb eines P2P-Netzwerks erfordert Ressourcen: Zeit, Strom, Rechenleistung und Internet-Datenvolumen. Die Anreizgestaltung der teilnehmenden Brokerknoten ist in Abschnitt 2.4.7 beschrieben.
- DATAcoin ist das Mittel der Kompensation zwischen Produzenten und Konsumenten. Mit anderen Worten: es implementiert einen Monetarisierungsmechanismus für Datenproduzenten. Dies ist ein Anreiz für Datenverkäufer, sich einzubringen und die Community zum Nutzen aller zu vergrößern.
- DATAcoin ist die Grundlage für Karma, die Reputationsmetrik in der Gemeinschaft der Datenproduzenten, Datenkonsumenten und Message- Broker. Parteien gewinnen Karma aus DATAcoin-Transaktionen: Daten veröffentlichen, Daten konsumieren und Brokerknoten betreiben, die das Gerüst für das Netzwerk bilden. Datenproduzenten gewinnen Karma, wenn von ihnen veröffentlichte Events an Abonnenten ausgeliefert werden. Abonnenten verdienen Karma, indem sie Events empfangen oder abrufen. Brokerknoten verdienen Karma für ihre Hilfe bei der Datenbereitstellung und Persistenz. Die Buchhaltung ist einfach: Die Menge an neuem Karma entspricht der Menge an DATAcoin, die ausgetauscht wird. Der Unterschied besteht darin, dass das Karma zerfällt und irgendwann seine Gültigkeit verliert, während der Tokensaldo nicht verfällt.

DATAcoin ist als ERC20-Token auf Ethereum implementiert. Der zum Token gehörende intelligente Vertrag hält die DATAcoin-Bilanz aufrecht und ermöglicht es, dass Zahlungen vertrauensvoll und sicher abgewickelt werden. Die Einhaltung des ERC20-Standards gewährleistet die Kompatibilität mit Wallets und anderen Token.

DATAcoin wird in einem Token Generating Event (TGE) erstellt, das derzeit für September 2017 geplant ist. Die Einzelheiten, die Bedingungen und der genaue Zeitplan werden zu einem späteren Zeitpunkt bekannt gegeben.

#### 4. Aktueller Status

Es gibt bereits eine hochmoderne Plattform für die Erstellung von Datenpipelines, Visualisierungen und blockkettenunabhängigen Computing-Mikrodienstleistungen. Die Plattform bietet einen funktionalen Ausgangspunkt, aber um eine vollständige Dezentralisierung zu erreichen, muss sie umgerüstet werden, um in einem dezentralen Container laufen und die neue Streamr-Netzwerkschicht für den Nachrichtentransport nutzen zu können.

Wir fangen nicht bei Null an. Es gibt eine funktionale und hochentwickelte Plattform für die Erstellung von Datenpipelines und Visualisierungen sowie für blockkettenunabhängige Rechenleistung und auf Ethereum basierende intelligente Verträge. Die Software ist für die Cloud-Umgebung im Hinblick auf Skalierbarkeit, Integrativität und eine geringe Fehlertoleranz ausgelegt. Große Data-Frameworks wie Kafka und Cassandra werden für die Datenverwaltung und das Messaging eingesetzt. Die Streamr-Plattform wurde im Februar 2017 in EDCON live demonstriert und seitdem in verschiedenen Blockketten-Meetups gezeigt.

Was den Stammbaum betrifft, so haben wir die erste Version der Software für den Eigengebrauch im algorithmischen Hochfrequenzhandel vor bald fünf Jahren erstellt. Die Führungsriege von Streamr hat einen Finanz- Hintergrund und alle Mitglieder waren entweder im Bereich Quantitative Finance tätig, als Entwickler von Handelssystemen, als Stat-Arb-Trader und manche Teammitglieder waren sogar alles zusammen. Quantitative Finance ist ein Bereich, in dem die automatische Verarbeitung großer Mengen von Echtzeitdaten seit 10-15 Jahren Realität ist. Erst in den letzten Jahren findet diese Vorgehensweise und die dort eingesetzte Art von Werkzeugen und Plattformen ihren Weg in die Welt von IoT, IoE und jetzt auch in den Blockkettenraum.

Die aktuelle Plattform ist funktional, skalierbar und wird bereits von Firmenkunden live genutzt. Die meisten der Komponenten lassen sich jedoch nicht direkt in die neue Welt übertragen. Der Speicherplatz muss dezentralisiert werden, die Messaging- und Publish/Subscribe-Funktionalität sowie Mechanismen zur Datenmonetarisierung und -verschlüsselung müssen in die Transportschicht integriert werden, und das Peer-to-Peer-Netzwerk muss zusammen mit der Knotenkoordination und einem Anreizmechanismus eingerichtet werden. Die Roadmap, wie dies alles geschehen soll, wird im nächsten Abschnitt vorgestellt.

## 5. Roadmap

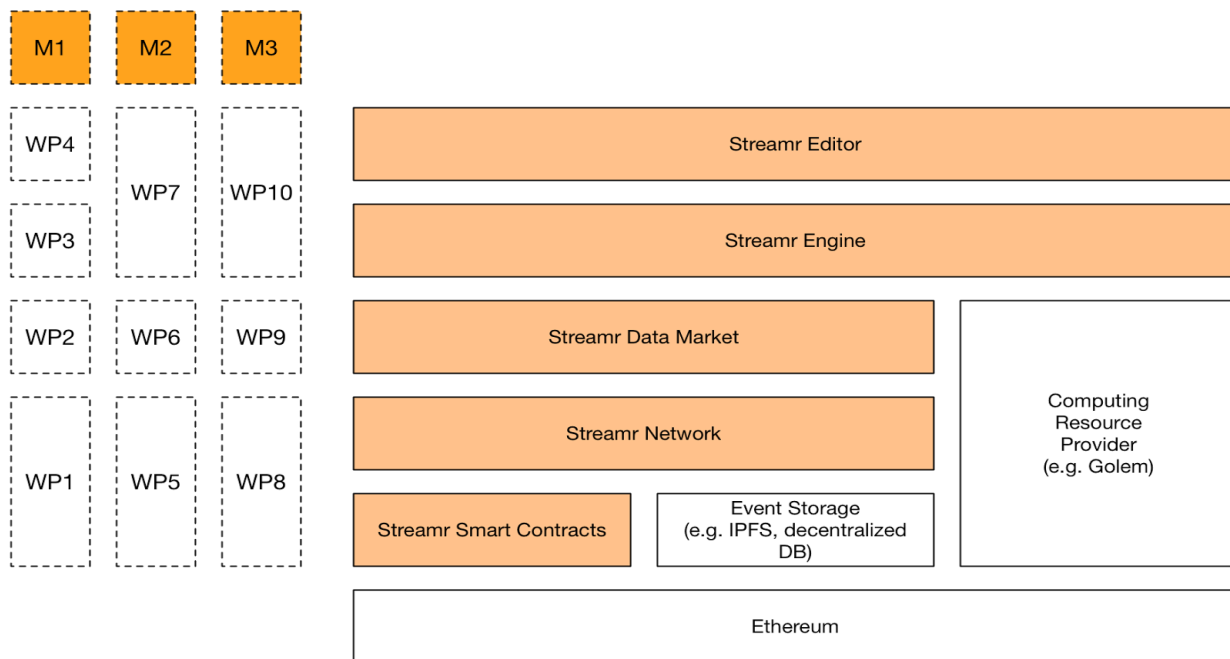


Abbildung 7. Forschungs- & Entwicklungs-Roadmap für das Streamr-Projekt.

Die Roadmap (Abbildung 7) gliedert sich in drei Meilensteine (M1-M3). Jeder Meilenstein fügt den Schichten des Technologiebaums (Streamr Stack) neue Funktionen hinzu. Jeder der drei Meilensteine besteht aus Arbeitspaketen (Working Package 1 – Working Package 10, nachfolgend 'WP' genannt). Das Abschließen aller WPs innerhalb eines Meilensteins schließt den Meilenstein ab. Jedes Paket hat einen speziellen Fokus auf bestimmte Schichten im Technologiebaum. Alle WPs in einem Meilenstein werden ungefähr gleichzeitig bearbeitet, aber die Arbeit an WPs in dem nächsten Meilenstein beginnt erst, wenn der aktuelle Meilenstein komplett abgeschlossen ist.

Am Ende jedes Meilensteins wird es auch eine Sicherheitsprüfung geben. Alle intelligenten Verträge werden geprüft, ebenso wie relevante Teile eines nicht intelligenten Vertragscodes, z.B. der Broker-Client selbst.

Wir haben uns aus den folgenden Gründen für den iterativen Full-Stack-Ansatz entschieden:

- Wir können der Community vom ersten Tag an etwas bieten, das funktioniert und für sie nutzbar ist.
- Wir beginnen mit einer bestehenden Technologiesammlung. Diese ist modular aufgebaut, sodass beliebige Ebenen an jeder Stelle ganz nach Bedarf nachgerüstet werden können.
- Konkrete und nützliche Ergebnisse werden unabhängig davon verwirklicht, welche Meilensteine wir erreichen. Die Community wird nicht mit einer halbfertigen Lösung im Stich gelassen, sondern es wird eine funktionale Technologie geben, die viele Einsatzmöglichkeiten im geschäftlichen Umfeld bietet.

## 5.1 Meilenstein 1

Meilenstein 1 sieht die Veröffentlichung der ersten Version des Datenlieferungsnetzwerks und der zugrunde liegenden intelligenten Verträge vor. Die Arbeit auf anderen Ebenen baut auf dem bereits erreichten auf und hat die vollständige Integration in das Ethereum-Ökosystem zum Ziel.

### WP1 (Netzwerk, Intelligente Verträge): Prototyp eines dezentralen Brokernetzwerks

- Erstellen des ersten Prototypen des dezentralen Brokerknotens
- Integration von libp2p
- secp256k1 ECDSA Signatur und Verifikation
- Eventprotokoll
- JSON- und Websocket-APIs
- Stream - Intelligenter Vertrag
- Stream Registry – Intelligenter Vertrag
- Netzwerk-Controller - Intelligenter Vertrag
- Partitionszuordnung
- Prüfsummen-Reporting
- Flood-Prävention durch Veröffentlichungs-/Abonnementgebühren
- Grundlegendes DATAcoin Prämiensystem

Die alte "Cloud"-Messaging-Ebene und das neue dezentrale Netzwerk werden für eine Weile nebeneinander existieren, bis der dezentrale Broker eine ausreichende Größe und Stabilität erreicht hat. Die

Engine und der Editor laufen mit dem alten Broker, bis die Netzwerkebene aktualisiert wird.

### WP2 (Marktplatz): Grundfunktionen des Datenmarktplatzes

- Auffinden von öffentlich zugänglichen Streams, Erstkategorisierung und Suche
- Stream-Produkt-Implementierung, käufliche Artikel, die den Zugang zu einer Reihe von Streams im Austausch gegen DATAcoins ermöglichen.
- Benutzer können Stream-Produkte erstellen und auf dem Marktplatz anbieten.
- Bestücken Sie den Marktplatz mit Echtzeit-Datenströmen aus verschiedenen Branchen, wie z.B. den folgenden:

- Finanzmarktdaten: Aktienkurse, Optionspreise, Rohstoffpreise, Kryptowährungskurse, Fiat-Wechselkurse, Handelsvolumen und mehr.
  - Social-Media-Daten: Twitter, Instagram, Facebook, reddit, flickr, etc.
  - Verkehrsdaten: Abflüge, Ankünfte, Geolokalisierung, Geschwindigkeit, Richtungsdaten von Flugzeugen, Schiffen, Zügen und dem Nahverkehr.
  - Wetterdaten: Temperatur, Niederschlag, Luftfeuchtigkeit, Bewölkung, Windgeschwindigkeit, aktuelle sowie auch prognostizierte Werte.
- Die erste Version des Marktplatzes wird einige zentralisierte "Trainingsfunktionen" haben, um die Zugriffsverwaltung zu vereinfachen und die Zentralisierung/Dezentralisierung des Produktionsnetzwerks zu spiegeln.

### **WP3 (Engine): Die Engine läuft mit Ethereum**

- Streamr-Web3-Bridge zur Unterstützung von Ethereum-Interaktionen von Streamr
- Ausgabe von intelligenten Verträgen, ABI-Support für vorinstallierte Verträge
- Lokale und transaktionale Funktionsaufrufe
- Eventbeobachtung
- Key- und Account-Management
- Unterstützung für verschiedene Test- und Hauptnetze
- Signaturen und Signaturprüfungen
- Herstellen der Kompatibilität der Engine mit Golem oder anderen dezentralen Anbietern für Rechenleistung

### **WP4 (Editor): Nahtloser Zugriff auf Rechenleistung, sowohl auf der Blockkette als auch blockkettenunabhängig**

- Verbesserung des visuellen Editors, um alle ethereumbezogenen Funktionen der Engine vollständig zu unterstützen.
- Integrierter Solidity-Editor zum Schreiben von benutzerdefinierten, intelligenten Vertragsmodulen
- Eine integrierte Auswahl an intelligenten Vertragsvorlagen für die gängigsten Anwendungsfälle (Zahlungen, Wetten, SLA-Überwachung, Prognosen usw.) und die Möglichkeit, Datenströme aus der realen Welt einfach auf diese Vorlagen anzuwenden.
- Überarbeitung des UI/UX des Editors sowie der dazugehörigen Webapplikation

## **5.2 Meilenstein 2**

In diesem Meilenstein veröffentlichen wir die erste Version des Datenmarktplatzes, zusammen mit den Funktionen, die er in der darunter liegenden Netzwerkebene benötigt.

### **WP5 (Netzwerk, Intelligente Verträge): Datenmonetarisierung und Verschlüsselung**

- Erste stabile Version
- Grundlegende Verschlüsselung
- Intelligente Verträge für Datenlizenzen
- Intelligente Verträge als Abonnementziel
- Weitere Signaturmethoden, z.B. basierend auf X.509, SGX, etc.

- Basisspeicher wahlweise in dezentralem Blocklager oder dezentraler Datenbank
- Implementierung und Nutzung von Karma
- Stresstests, Optimierung der Skalierbarkeit

#### **WP6 (Datenmarktplatz): Vollständig dezentralisierter Datenmarktplatz**

- Die "Stützräder" aus der anfänglichen Marktplatz-Implementierung in M1 werden nun entfernt, um eine vollständige Dezentralisierung zu erreichen; dazu gehören auch Datenlizenzen, die als intelligente Verträge auf der Blockkette modelliert wurden und für die Genehmigung und Zugriffskontrolle verwendet werden
- Urversion eines Schlüsselverteilungsmechanismus für erlaubnisgesteuerte Streaminhalte in einem öffentlichen Netzwerk
- Überprüfung der Identität des Verkäufers
- Mechanismus zur Bildung von Verkäuferbewertungen

#### **WP7 (Engine, Editor): Dezentralisierung erreichen**

- Migration auf die neue Netzwerkebene
- Einsatz auf Golem oder einem anderen Containeranbieter
- Fehlerrobustheit und Fehlerbehebung in dezentraler Umgebung

### **5.3 Meilenstein 3**

Das Ziel von Meilenstein 3 ist es, die volle Streamr-Vision zu verwirklichen. Die F&E von Meilenstein 3 wird sich im Laufe der Zeit ändern, da die bisherigen Meilensteine verschiedene Ideen hervorrufen und die Community neue Funktionen einfordern wird. Meilenstein 3 wird auch erhebliche Marketinganstrengungen beinhalten, um die Akzeptanz des Technologiebaums zu erhöhen.

#### **WP8 (Netzwerk): Erweitertes Routing, Standortverschleierung**

- Standortverschleierung
- Multicast-Verschlüsselung
- Stresstests, Optimierung der Skalierbarkeit
- Arbeit an allen Problemen, die bei groß angelegten Implementierungen auftreten
- Arbeit an Funktionen, die von der Community gewünscht werden
- Arbeit an der Integration mit neuen Plattformen

#### **WP9 (Datenmarkt): Bildung einer Community**

- Hinzufügen weiterer Streams zum Datenmarktplatz
- Stream-Wunschliste, Prämienprogramm, um das Wachstum der Plattform zu beschleunigen
- Verbesserung des Reputationsmechanismus
- Verstärkte Bemühungen um den Ausbau der Community

#### **WP10 (Engine, Editor): Finale Schritte zur Produkterstellung**



- Verbesserung des UI und UX der Werkzeuge
- Onboarding, Tutorial-Videos, Hilfsmaterialien
- Hinzufügen der Integrativität mit relevanten Plattformen im Blockkettenbereich, IoT, AI oder anderen Bereichen.

## 6. Projektmanagement-Team



**Henri Pihkala, M.Sc.**

Henri ist ein Software-Ingenieur, ein Serien-Unternehmer und ehemaliger algorithmischer Händler. Er leitete die Entwicklung von zwei Hochfrequenz-Algorithmus-Handelsplattformen. Er zeichnet für den Entwurf sowie die Entwicklung der verteilten Streamr Cloud Analytics Plattform verantwortlich. Henri beschäftigt sich leidenschaftlich gern mit den Themen komplexe Architektur, Skalierbarkeit, Benutzerfreundlichkeit und Blockketten.



**Risto Karjalainen, Dr. Phil.**

Risto ist ein Datenwissenschaftler und Finanzexperte mit einem Doktor der Philosophie von der Wharton School an der Universität von Pennsylvania. Er ist ein Spezialist im Bereich Quantitative Finance und kann eine internationale Karriere im Bereich des automatisierten, systematischen Tradings sowie im Asset-Management vorweisen. Risto's Interesse erstreckt sich über die Bereiche Echtzeit-Computing, maschinelles Lernen, evolutionäre Algorithmen, Verhaltensökonomik, Blockketten und finanzielle Technologien im Allgemeinen.



**Nikke Nylund, B.Sc.**

Nikke ist ein ehemaliger Stratege für algorithmisches Low-Latency-Trading. Er hat mehr als 20 Jahre Erfahrung als Manager, Unternehmer, Gründer und/oder Investor im Bereich der ICT- und Tech-Unternehmen und kann zahlreiche erfolgreiche Exits aufweisen. Nikke hat den akademischen Grad des BSc in den Bereichen Finanzierung und Unternehmertum von der Helsinki School of Economics inne.



**Michael Malka, M.Sc.**

Michael ist ein Unternehmer und Technologieenthusiast mit 20-jähriger Erfahrung in verschiedenen Rollen, vom Softwareentwickler bis zum Vorstandsvorsitzenden. Er war bei diversen Softwareprojekten in verschiedenen Sektoren eingebunden – von Startups über Banken bis zu Telekommunikationsunternehmen. Michael studierte Computerwissenschaften an der Universität von Helsinki bevor er sein erstes Softwareunternehmen gründete.

## 7. Fazit

Dieses Whitepaper beschreibt unsere Vision eines robusten, nativ-dezentralen Grundgerüsts aus Echtzeit-Daten für dezentrale Anwendungen. Wir gehen davon aus, dass die Kombination aus einem Echtzeit-Datenmarktplatz und einer Datenpipeline für die Entwickler von intelligenten Verträgen auf Ethereumbasis und das gesamte Dapp-Ökosystem von entscheidender Bedeutung sein wird. Unser Ziel ist es, einen gut durchdachten und professionell implementierten Technologiebaum zu entwickeln, der den zukünftigen Bedürfnissen unserer Zielgruppe gerecht wird und unaufhaltsame Daten für unaufhaltsame Apps liefert.

Unser Technologiebaum ist mehrschichtig und modular aufgebaut und basiert auf einer dezentralen Transportebene. Es gibt ein Peer-to-Peer-Netzwerk, das aus anreizgesteuerten Brokerknoten besteht. Das Netzwerk verfügt über einen Publish/Subscribe-Mechanismus und unterstützt die dezentrale Speicherung von verschlüsselten Events. Der Durchsatz skaliert linear mit der Anzahl der teilnehmenden Knoten, und das Netzwerk kann Millionen von Events pro Sekunde verarbeiten.

Das Daten-Grundgerüst ist ein idealer Katalysator für die M2M-Wirtschaft, in der autonome Maschinen, Bots und Roboter winzige Datenstücke kaufen oder verkaufen. Die Idee<sup>18</sup> dahinter ist, dass Maschinen Ressourcen wie Speicherplatz, Verarbeitungskapazität, Kommunikationsbandbreite usw. tauschen. Wir glauben, dass die Verwendung von DATAcoins zu viel niedrigeren Transaktionskosten führt als der reine Tauschhandel.

Streamr ist Teil der Computing-Revolution, bei der monolithische Lösungen durch dezentrale Computing-Alternativen ersetzt werden. Im Bereich des verteilten Computing ersetzt Golem die Azure Virtual Machines. Im Bereich Blockspeicher ersetzen IPFS, Storj und andere Azure Blob Storage. In Bereich der Datenpipelines und des Messaging ersetzt Streamr zentrale Messaging- und Eventverarbeitungsplattformen wie Azure EventHub und Azure Stream Analytics. Es findet ein Machttransfer von Konzernen und Unternehmen zu einzelnen Bürgern, autonomen Agenten, Apps und Maschinen statt, der zu Verbesserungen in den Bereichen Privatsphäre, Effizienz, Belastbarkeit und Fehlerrobustheit sowie letztendlich zu einem höheren Wohlstand für die Stammgäste der vernetzten Gesellschaft führt

---

<sup>18</sup>Alex Puig: "How the Blockchain could enable the Machine Economy (M2M)", 11. Januar 2016 (<https://www.linkedin.com/pulse/how-blockchain-could-enable-machine-economy-m2m-alex-puig>)