

# Introduction

[@markbmilton](#) and I have been experimenting with the prospect of creating a money “streaming” protocol, as discussed in this [keynote talk](#) by Andreas Antonopoulos. We bashed our heads against the wall trying to design a feasible solution on LN, Raiden and other state channels, eventually landing on Plasma. We have seen the light at the end of the tunnel - for now, at least!

## Project Goals

- Standardise continuous value transfers between peers, which we shall further refer to as “streams”
- Enable the usage of ERC20 tokens, with a focus on stable coins such as DAI
- Achieve fast finality and low fees when interacting with the ledger
- Reduce reliance on human promises (Ricardian contracts) and increase awareness about “blockchain as a court” (Plasma child-chain with streams in our case)

## Rationale

It is counter-intuitive to imagine financial commitments over non-trivial periods of time as not being set at fixed intervals (such as a month or a year). And rightly so! Fixed interval payments alleviate [mental transaction costs](#). However, if we imagine the time factor of recurring payments as an arbitrary variable, a series of unexpected benefits can be observed.

1. Increased flexibility: if the payment isn't taken once per month, rather once per hour, or minute, or even less, customers are afforded increased consumption flexibility.
2. Reduced counterparty risk: in situations where pricing is contingent on time (e.g. consultations), both parties would benefit from running a stream. In lieu of creating counterparty risk by pre/post paying for the consultation, the consultant knows they are getting paid for every time unit elapsed. Similarly, the customer doesn't have to pay upfront and can cancel the stream immediately based on their subjective view of the consultant (expert or charlatan).

There are many examples where granular recurring payments would align the incentives of all parties involved, but we haven't included them for brevity.

## Plasmatic Approach

The key question in our research was: how do we design an ecosystem where money flows continuously, i.e. it is a function of time, but that does not require users to sign a transaction every interval  $i$

(which should approach a very low number)? We concluded that we could reach logarithmic scalability by crafting a blockchain which understands *some* transactions as actions of starting and closing a stream. This blockchain would have its own consensus mechanism and use PoS as its Sybil protection mechanism, but it would periodically publish updates to a Plasma smart contract to inherit Ethereum's security.

## How It Works

Below is the general sequence of steps used for setting up a stream between Alice, a payer, and Bob, a payee. Funds can only flow from Alice to Bob, but Bob can also propose state updates. The following heuristics assume a Plasma Cash model.

1. At time  $s$

, Alice and Bob agree on a price  $p$

, payment interval  $i$

and stream closing time  $c$

1. Alice deposits funds worth:

$p * ((c - s) / i)$

1. Alice now controls a coin  $cid$

representing her funds on the root chain

1. Alice and Bob initiate the stream by creating a pseudo-escrow account

2. Alice and Bob can optionally mutate the state to  $p'$

,  $i'$

and  $c'$

and later hand over a cryptographic proof which redeems:

$$p' * ((c' - s) / i')$$

1. The stream is closed by Alice, before or when block  $c$

is generated, or by Bob, only when or after block  $c$

is generated

Importantly, Alice can close the stream at any time before block  $c$

. If she does so, the protocol performs an atomic swap by making Bob give Alice her chargeback in the form of a Plasma coin (it's assumed Bob has additional funding on the root chain if he doesn't have an exact denomination on the side chain). Otherwise, the stream is closed normally and Bob receives  $cid$

without any further obligations. Here's a visual representation:

## Notes

One might draw similarities between [ERC948](#) and Chronos, but the fundamentals are different. ERC948 focuses on recurring monthly payments (going lower would not be feasible to the accrual of fees), while Chronos aims to be much more flexible and target any peer-to-peer interactions, not just customer-business relationships.

We are keeping abreast of the latest discussions on [Plasma Debit](#), the [viability](#) of smart contracts on Plasma and the [ideation](#) on state channels + Plasma Cash. The Chronos implementation currently being build will only have a single Plasma operator and the minimum viable procedures applied.

The goals of this post are to let us share our ideas as well as provide a non-exhaustive introduction to a protocol which enables money streaming. The assumptions we made, along with encountered bottlenecks and in-depth technical details can be checked out in this draft white paper:

[chronos-white-paper.pdf](#)

We're standing on the shoulders of giants, so we'd like to address huge kudos to all people involved in researching Plasma, state channels and Ethereum itself! The quality of the stuff posted here is insane.

Paul Berg