The information in this document is still in **draft** mode, and **CONFIDENTIAL**. Please do not share the following information without prior approval from us.

Whitepaper

Decentralizing, restructuring, and optimizing the way we transfers goods.

Version 1g (limited release)

Israel Levin, Oren Gampel

Abstract

The ability to send, receive and transport goods is a fundamental part of today's economy, society, and way of life. The business of parcel delivery fuels an ever growing industry with an annual revenue of $300 billion.

However, the current, centralized way of handling deliveries is both limiting and wasteful, giving rise to problems so inherent that they are mostly taken for granted (the “Last Mile” problem being a glaring example). Overcoming these problems will not only improve deliveries as we know them, but will open the door to new classes of deliveries and an untapped variety of customers, providers, and use-cases, creating more opportunities in this already huge market even further.

PaKeT is using blockchain technology to disrupt the industry, enabling a win-win solution for the entire ecosystem: empowering local couriers, reducing operational costs for huge global shipping companies, and enabling faster, cheaper routes for the end user.

A decentralized, open market of deliveries in which anyone can participate — be them individuals or organizations, professional or casual, dedicated or opportunistic — will mitigate, reduce, and in some cases completely eliminate systemic inefficiencies that are practically unavoidable in the current, existing delivery market. This will result in the creation of faster, cheaper and more effective ways of transporting goods

To achieve this worthwhile goal we are creating a multi-layered cryptographically secure protocol for the delivery of anything to anywhere by anyone in a highly efficient manner. A manner that is not only cheaper and faster than current delivery methods but is also safer, more accountable, more diverse and completely transparent. A protocol that will enable smarter and more efficient collaboration between the folks who can move stuff and the folks who want stuff moved.

Content Of This Paper

In this paper we describe a protocol for establishing trust and cooperation between multiple parties regarding the safe and timely delivery of goods. We start with a brief technological introduction, giving a taste of the protocol's inherent value, and only then dive into the technical description and explore the five layers of the protocol:

- L0 - the decentralized consensus layer which establishes trust between the different parties
- L1 - the cryptographic token and the smart contracts which govern its behaviour
- L2 - the routing layer which matches the capacity and cost of couriers with the requirements of senders and recipients while providing a detailed and highly contextualized view into the supply and demand of the network
- L3 - the user layer which turns all these abilities into an accesible market
- L4 - the organizational layers which proposes organizations and and services which can thrive in this ecosystem and enrich it

We will examine user stories and make extensive use of sequence diagrams. In part to illustrate the workings of the protocol — its strengths and its flexibility — but also to make concrete suggestions as to different types of entities, both individuals and organizations, that can gain from participating in the network while increasing its ability to deliver packages quickly, cheaply and effectively.

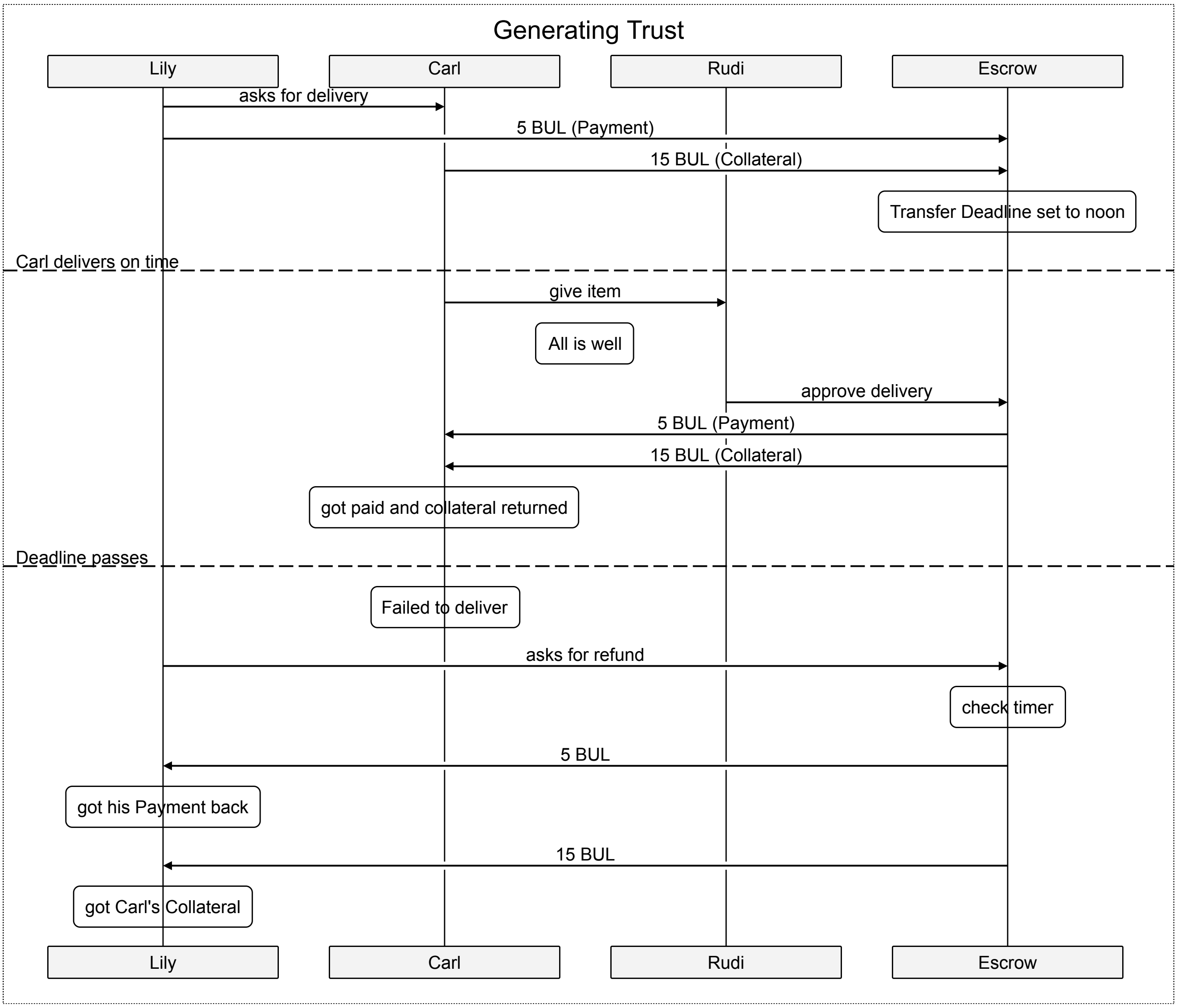
Technical Overview

The fundamental problem with decentralizing deliveries is that of establishing trust. Specifically, trust between the people who are willing to pay in order to have their goods transported (we call them Launchers) and the people who are willing to transport said goods in exchange for said Payments (we call those Couriers).

We offer a mechanism for establishing this trust using a cryptographic token which we call BUL. Our protocol ensures that when Lily (the Launcher of the delivery) sends an item to Rudy (the Recipient) by giving it to Carl (the Courier), her payment is securely held in virtual escrow until Rudi confirms that the package was received, or until a predefined Transfer Deadline passes. It ensures that only Rudy can approve the receipt of the package and that only with this approval will Carl get paid. We call this a Proven Delivery. It also ensures that if Rudi does not receive the package before the deadline passes, Lily's payment is refunded in full.

The protocol further allows Lily to require that Carl commit a Collateral which is likewise held in virtual escrow. This is a form of insurance: if Rudy does not receive the package by the Transfer Deadline, Lily will not only get her payment back, she will also get Carl's Collateral as compensation. The size of the Collateral generally reflects the value of the item, so if Lily wishes to send something very expensive she is likely to demand a higher Collateral. And Carl can accept, or refuse, or even ask for a higher Payment since he is performing a high-risk delivery.

The system itself makes no attempt to determine prices as everything is left to the market economy — i.e. whatever Lily and Carl agree to.



Relay And Route

The same protocol allows Carl to take the package only a part of the way, and then pass it on to Neo (a new Courier) in return for a part of the Payment that Lily originally promised. The protocol ensures that the Payment is divided correctly upon Proven Delivery and that Neo can cover Carl's Collateral out of his own pocket, at which point Neo becomes responsible for the delivery, or in our terms: the Custodian of the package.

Neo can further transfer Custodianship over the package to another Courier, and another. As many as it takes to create a solid route between Lily and Rudi. Lily does not have to make any allowances for these relays — her required Collateral is always taken care of by the current Custodian, and market forces drive the delivery in the economically optimal path.

It is this last characteristic, the natural composition of routes in accordance with market forces, combined with a decentralized market - open to anyone who wishes to participate in it and transparent to anyone who wishes to examine it - which result in the optimization of the Route according to whatever conditions Lily sets (speed, price, reputation, insurance, or even the availability of refrigerated containers), and the optimization of the system as a whole.

Going Beyond

There are many organizations that can play a role in this new economy. Insurance companies, for example, committing Collateral on behalf of Couriers. Or Courier associations, which are a form of decentralized shipping companies, sharing Collateral pools and reputation.

Another form of Courier are existing delivery services, especially the incumbent giants of the industry. Such companies can't match the speed and efficiency of a bike riding New-York delivery guy, but they may be much better at shipping freight containers from London to New York. Consider an overseas shipment: Carl can pick up the package from Lily's home in London and use an international shipping company to send it to Neo in New York. Once the package arrives, Neo will ride his bicycle over to the post office, pick up the package and deliver it to Lily.

We also foresee an emergence of hubs on many of the more active areas of the network — physical locations to which packages are brought and from which they are picked up, either by Relay Couriers or by the Recipients themselves. These too are likely to form within existing establishments such as stores and gas stations.

There are many additional concepts, all built on top of the above, which can increase the utilization of the ecosystem, and we have some great ideas on how to develop the required software, how to bootstrap the ecosystem and how to help it grow organically. For further details, just read on.

The Protocol

The PaKeT protocol is a set of procedures that establish incentive and trust between multiple parties regarding the safe, secure and timely delivery of goods.

Layers

The protocol is composed of layers, with each layer providing utility to the layers on top of it. Layers are designed to be as modular and agnostic as possible so that one can always implement a different layer n on top of an existing layer n-1, and can even implement a whole different stack of layers (m-n to m-1) *underneath* an existing layer m. Also note that any application written over layer n is considered, ipso facto, an implementation of layer n+1.

In our design we try to find the golden path between two conflicting principles:

- Required functionality should be provided by the lowest possible layer.
- Each layer should directly provide as little functionality as possible.

- Layer 0 – Trust: the underlying cryptographic framework for achieving decentralized consensus.
- Layer 1 – Paket: transfer of goods between multiple parties with cryptographically assured payment and collateral.
- Layer 2 – Route: finding and establishing optimal routes for transfer.
- Layer 3 – User: simplifying human usage of the network to send packages, transport them and deliver them while creating value for themselves and the network.
- Layer 4 – Organization: unite participants in organizations that operate within and on top of the network while adding further value.

The authentication of entities throughout the layers is achieved using asymmetric cryptographic keypairs, which allows for the pegging of entities at every layer to the corresponding entities in any other layer. In other words, the same logical address identifies a participant in all layers.

Our design encourages the development of many competing alternatives to layers 2 and above. Such alternatives, as well as the competition between them, will serve to increase the utility of the network and its underlying token, which is defined in L1.

Important Notes Concerning Examples

- All the examples supplied, along with the sequence diagram, are “bare bones” examples provided specifically to shed light on the workings and abilities of the different layers.
- The user is never expected to use any of the layers directly, and user applications are expected to occupy layers 3 and 4.
- All the rates and sums described in the examples are arbitrary and used for illustration only. The actual value of BULs can only be determined by the real-world utility of the network.

Layer 0 – Trust

Layer 0 (L0) is the base of the protocol. It provides the layers above it with a tool for achieving consensus on the conditional transfer of funds between parties, as well as inspectable, immutable history of that consensus.

Elements

- Address – as a representation of a cryptographically verified entity (implemented as an asymmetric keypair)
- Blockchain – an open, immutable, decentralized ledger of transactions reflecting the system's consensus and state

Description

This layer provides the framework for writing, publishing, enacting and enforcing smart contracts that govern the transfer of funds, as represented by tokens, between addresses on the blockchain. It is based on decentralized and open ledger technology.

Implementation

While it is entirely feasible that we develop our own technology with our own blockchain – either merge-mined or standalone and probably based on or derived from an existing trusted technology – we believe there is already plenty of effort going into the creation of blockchain technologies that support smart contracts. We, therefore, choose to use an existing technology which lets us focus on solving less general problems. We are, of course, keeping our own implementation as agnostic as possible, so that we can adapt to the most fitting smart contract technology available.

We are currently looking into the following options:

- Ethereum – a separately mined blockchain which focuses on running smart contract code
- Rootstock – a merge-mined blockchain which extends Bitcoin's abilities to provide flexible smart contracts
- Cardano – a PoS based layered blockchain with clear seperation between computation and accounting
- Pure Bitcoin – somewhat harder to implement, but most trusted and tested

Layer 1 – Paket

Layer 1 (L1) establishes the transactional framework for the delivery of a Paket by a Courier incentivised by a Launcher and using a cryptographic token called a BUL for both Payment and Collateral.

Elements

- Token — a cryptographic token representing funds (ours is called BUL)
- Launcher — whoever sets the transfer up by committing funds to its Payment
- Courier — anyone who commits to delivering the Paket for a promised Payment by committing a Collateral
- Custodian — whoever currently has committed Collateral for the Paket. Usually, the Courier that currently holds the package, but Custodianship may be shared by multiple participants
- Payment — an amount of BULs promised by a Launcher to a Courier as compensation for his effort
- Collateral — an amount of BULs required by a Launcher from a Courier as insurance and as proof of commitment
- Transfer Deadline — the latest time in which a delivery can be made to the Launcher's satisfaction

Description

Layer 1 describes the delivery of a single Paket. It describes how Payment is promised and made, how Collateral (as a form of insurance) is handled, and how the Transfer Deadline which limits the delivery window is defined. It also describes a Relay, which enables both planned and opportunistic subdivision of the delivery Route between multiple Couriers.

Transfers

The transfer of a Paket from a Launcher to a Courier is logically governed by two agreements:

- The **Launcher** commits a certain amount of BULs **as Payment** for the delivery. These BULs leave his possession and are held in a virtual escrow mechanism. They will only be transferred to the Courier upon a successful Proven Delivery. If the delivery is not completed before the Transfer Deadline the Payment will be returned to the Launcher.
- The **Courier** commits a certain amount of BULs **as Collateral**. These BULs are likewise held in virtual escrow and will be returned to the Courier only upon successful delivery. If the package is not successfully delivered by the Transfer Deadline and in accordance with all requirements set by the Launcher these BULs will be given to the Launcher as compensation for the failed delivery.

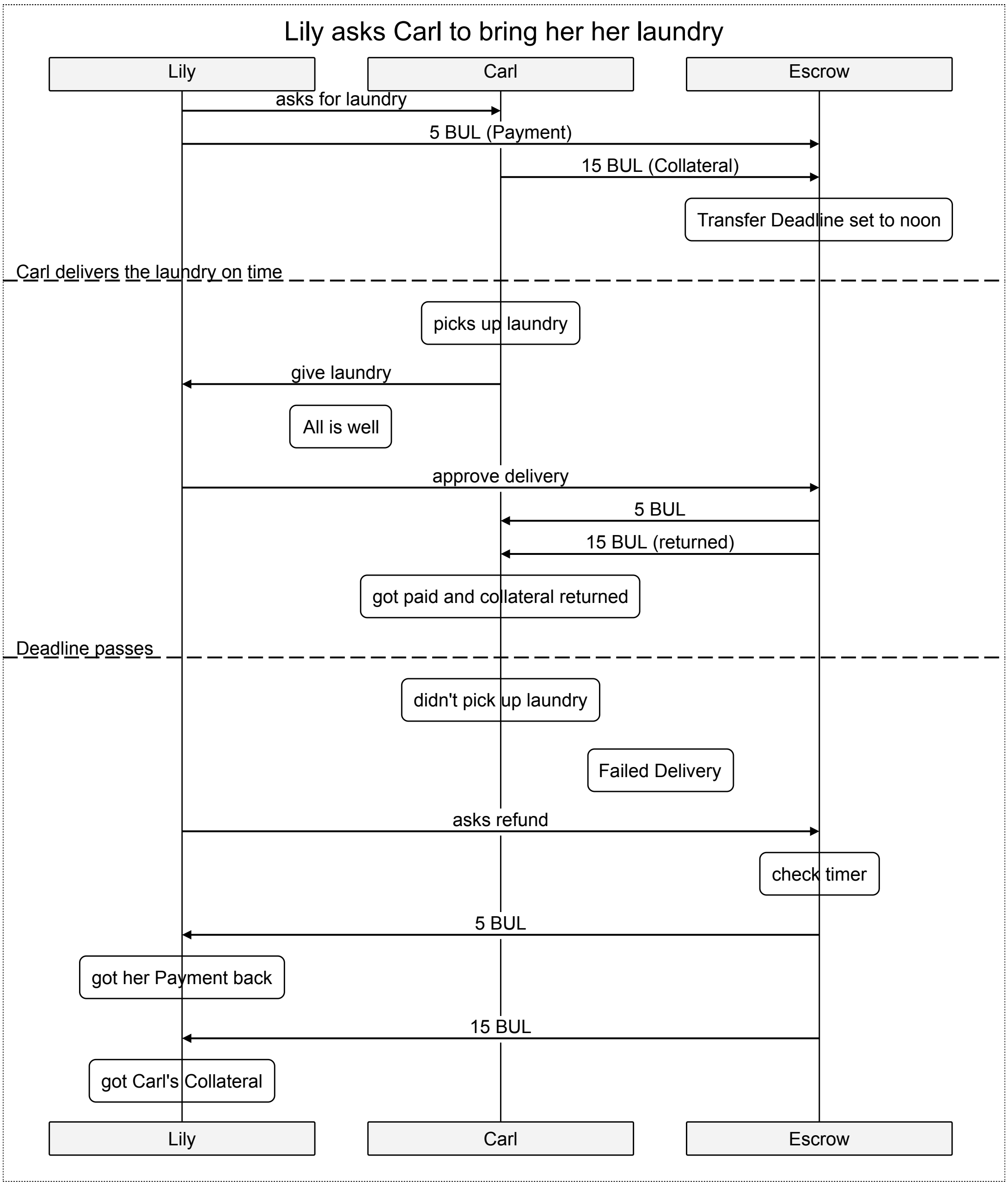
The actual implementation can split these agreements into multiple transactions or consolidate them into a single one, but once those two agreements are accepted and signed by Launcher and Courier and published down to L0, both parties have their assurances. Only when both are committed financially is the Courier given possession of the Paket and becomes its Custodian.

To define a successful, or Proven Delivery, an external Secret is generated by the original Launcher and shared with the final destination of the Paket (if they are not the same entity, which may often be the case). In a pure Bitcoin implementation, this Secret serves as a cryptographic hashlock on all related Payment transactions. In a Solidity based solution, we can make do with the Launcher's signature, which represents his private key (which can be considered a Secret generator).

The following example shows how the L1 protocol supports a simple delivery. It's again important to note that in the real world user applications are not expected to deal with L1 directly, but will use L2, or even more likely L3 and L4 to handle the details for them. The examples in this section are just that, examples provided for the sole purpose of clarification.

Example: Simple Delivery (Launcher, Courier)

Let's imagine Lily, our Launcher, wants her clothes delivered to her from the dry cleaner's. She contacts Carl and promises him a Payment of 5 BULs if he delivers them by noon, but demands a collateral of 15 BULs in case Carl fails to deliver them on time (or loses them altogether).



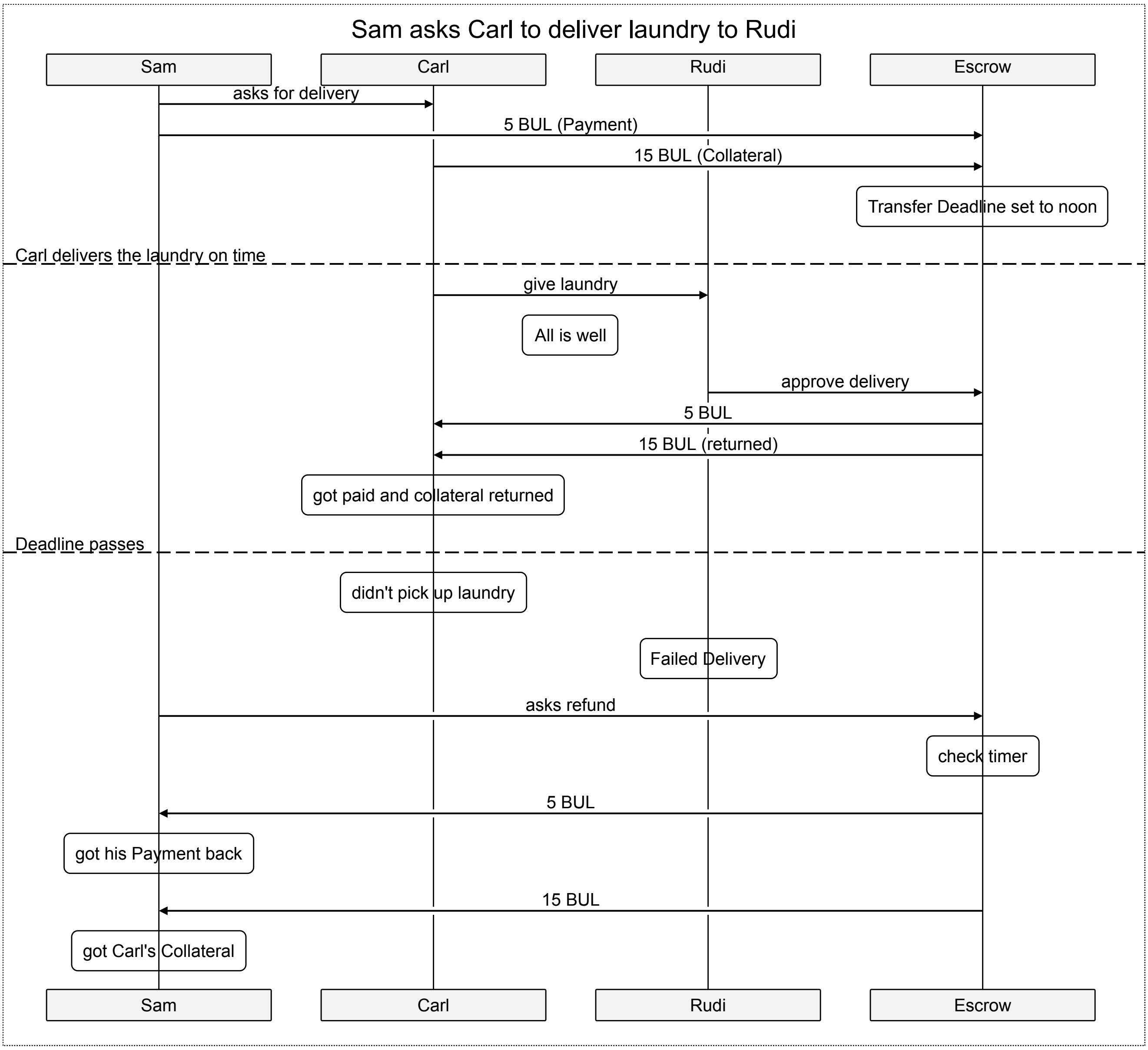
Collect

Although it makes perfect sense that the Payment will always come from the entity receiving the Paket — one pays in order to receive something one desires — in the world of shipping and delivery it often seems to be the other way around. We pay for stamps that let us cede possession of a package and have it delivered to a remote address.

Our protocol allows for such cases using the Collect delivery. In such a delivery the Launcher, the initiator of the delivery, is not the destination of the Paket. To clarify, we will call that Launcher a *Sender*. And while the Sender can't verify the delivery of the Paket (by virtue of not being there) he does know, or at least has the address of, someone who can — the Recipient. The simplest solution is usually for the Recipient to generate his own secret upon which the Launcher hinges the Payment. Another solution is for the Launcher to generate the secret and share it with the Recipient over a secure channel. In any event, the Sender, who is also the Launcher, places the Payment in the virtual escrow, as usual, with the difference of granting the Recipient allowance to transfer it to the Courier or Couriers.

Example: Collect Delivery (Sender, Courier, Recipient)

Let's imagine that our Sender is really the dry cleaner himself, Sam. And he wants Carl, or anyone else, to get some freshly cleaned clothes back to their rightful owner, Rudi, the Recipient. He deposits 5 BULs in escrow as Payment, stipulating that only Rudi can release them, which means that Rudi must have the secret. It is likely that Sam simply uses Rudi's known address for this stipulation (assuming a corresponding private key Rudi controls), but there are many other ways. Sam could pre-generate secrets for all his clients and print them on their receipts. Or maybe Rudi uses an app that generates secrets for exactly such cases. It's even possible that Sam calls Rudi and gives him a passphrase over the phone.



Relay

At any point along the way, the Courier may wish, instead of completing the delivery on his own, to Relay the Paket to another Courier. He is likely to promise the new Courier a smaller payment than what he was originally promised in order for him to make a profit. He is also likely to demand that the new Courier covers the Collateral.

In short, a Relay means that the Courier becomes a new Launcher and that a new participant acts as a new Courier, with the former Courier now offering Payment and demanding Collateral from the new one, thus passing along Custodianship of the Paket. And this can happen any number of times until the paket reaches its destination.

While L1 is deliberately flexible, higher layers can use this Relaying Payment/Collateral mechanism to enforce integrity and cohesion by:

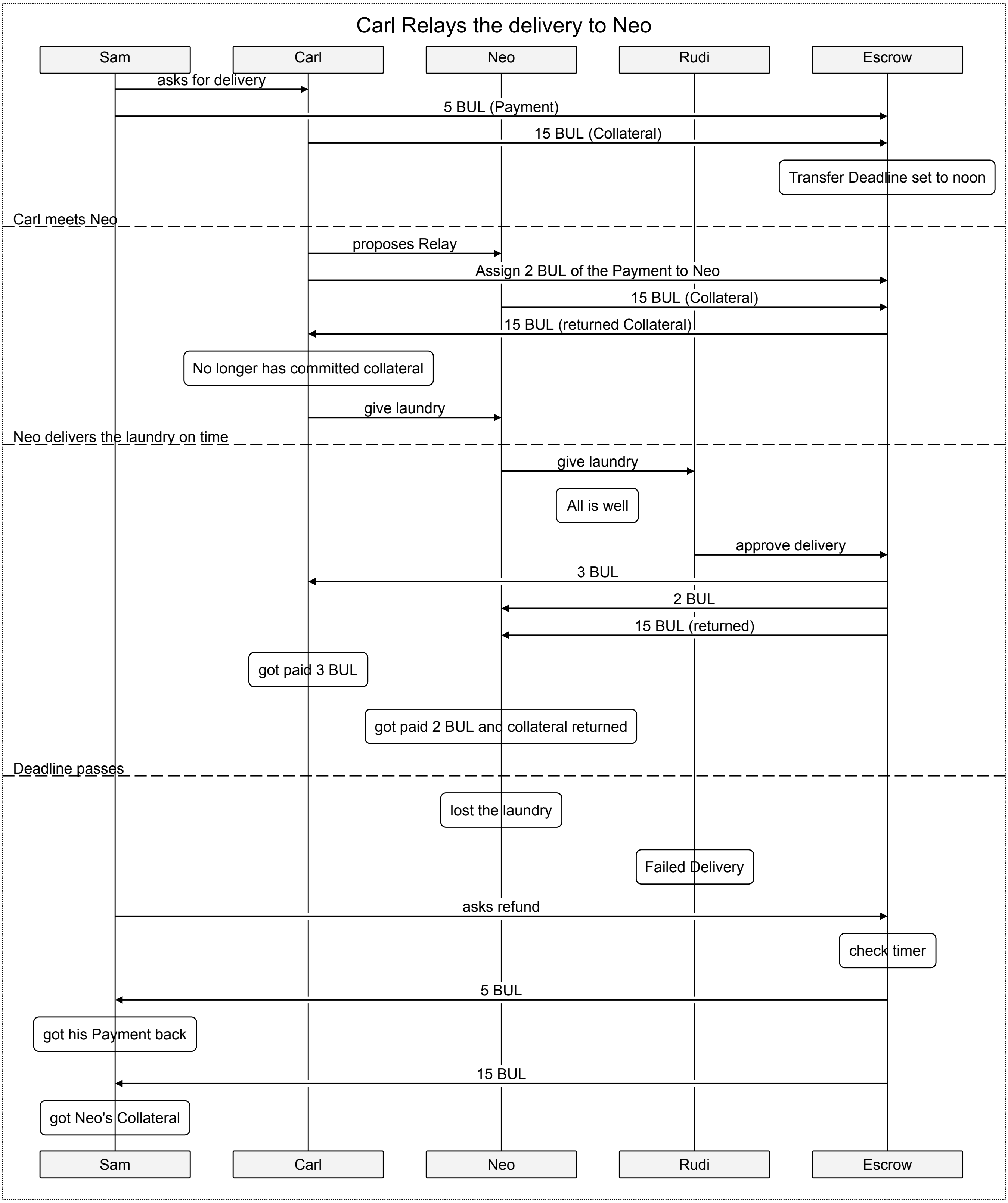
- Ensuring the new Courier completely covers the old Courier's Collateral with his own BULs.
- Ensuring the new Launcher covers the new and reduced Payment with the BULs originally committed as Payment by the old Launcher.
- Conditioning the new Payment transactions on the same condition - the secret that proves the delivery of the Paket to the Recipient.

In implementations that do not allow a complex scripting language, such as a pure Bitcoin implementation, it is important to remember that all Payments are hinged on the same external secret. Once the final Courier in the chain delivers the Paket to the Recipient and uses the secret to withdraw his Payment, the secret is uncovered over the blockchain and all BULs flow to their rightful owners. This can theoretically include Payments made on multiple blockchains.

Example: Collect Delivery With Single Relay (Sender, Courier, New Courier, Recipient)

Let's imagine that on his way to Rudi's house Carl gets a bit tired. He still has plenty of time, so he sits down for a cup of coffee at a local cafe where his friend Neo shows up. Carl remembers that Neo lives across the street from Rudi, so he quickly initiates a Relay, offering Neo (the new Courier) 5 BULs for getting the clothes to Rudi before noon, and demanding that Neo cover the existing Collateral of 15 BULs. Neo happily agrees and commits 15 BULs to the virtual Escrow, which covers Carl's Collateral. Carl immediately gets his 15 BULs back.

Note that this is equivalent to Neo giving Carl 15 BULs in exchange for the assurance that Carl's originally deposited Collateral be given to Neo in case of a successful delivery. And indeed, in many implementations, the two descriptions are indistinguishable as BULs are completely fungible and as both transfers (from Neo to the escrow and from the escrow to Carl) happen atomically on the same layer 0 transaction.



Raising Payment

It is always possible to raise the amount of BULs offered as Payment, even when a Paket is in transit and between relays. This can be handy in many situations involving opportunistic routes. It can be used to initially offer an optimistically lower price and then slowly raise it, when the priority of the delivery changes, when the destination changes (e.g. when you have to get out of town and want the delivery to follow you) or when there is a change in any other condition (e.g. it started raining and you feel like adding a tip).

Generally, this will be done by either the Recipient or by any of the prior Launchers – all of them have a clear interest in the delivery taking place - and the extra Payment is usually promised to the current Custodian of the Paket – whom the additional promised BULs further incentivize to complete the delivery or to increase the Payment he is willing to offer the next Courier.

However, the Payment for the Proven Delivery of a Paket can be committed by anyone and promised to anyone. Any interested party can deposit more BULs into the virtual escrow connected with the Paket and give any other party control over them when the delivery is successfully completed. As always, if the delivery fails the BULs are returned to their original owners.

Example: Raising Payment On A Collect Delivery With A Single Relay (Sender, Courier, New Courier, Recipient)

Let's imagine that while Rudi is waiting for his clothes he remembers he has an important meeting on the other side of town. And not only does Rudi still want his clothes, but he is also worried about his reputation as a reliable Recipient, a reputation he has been building for a few months and which is the main reason Couriers are willing to give him such great prices.

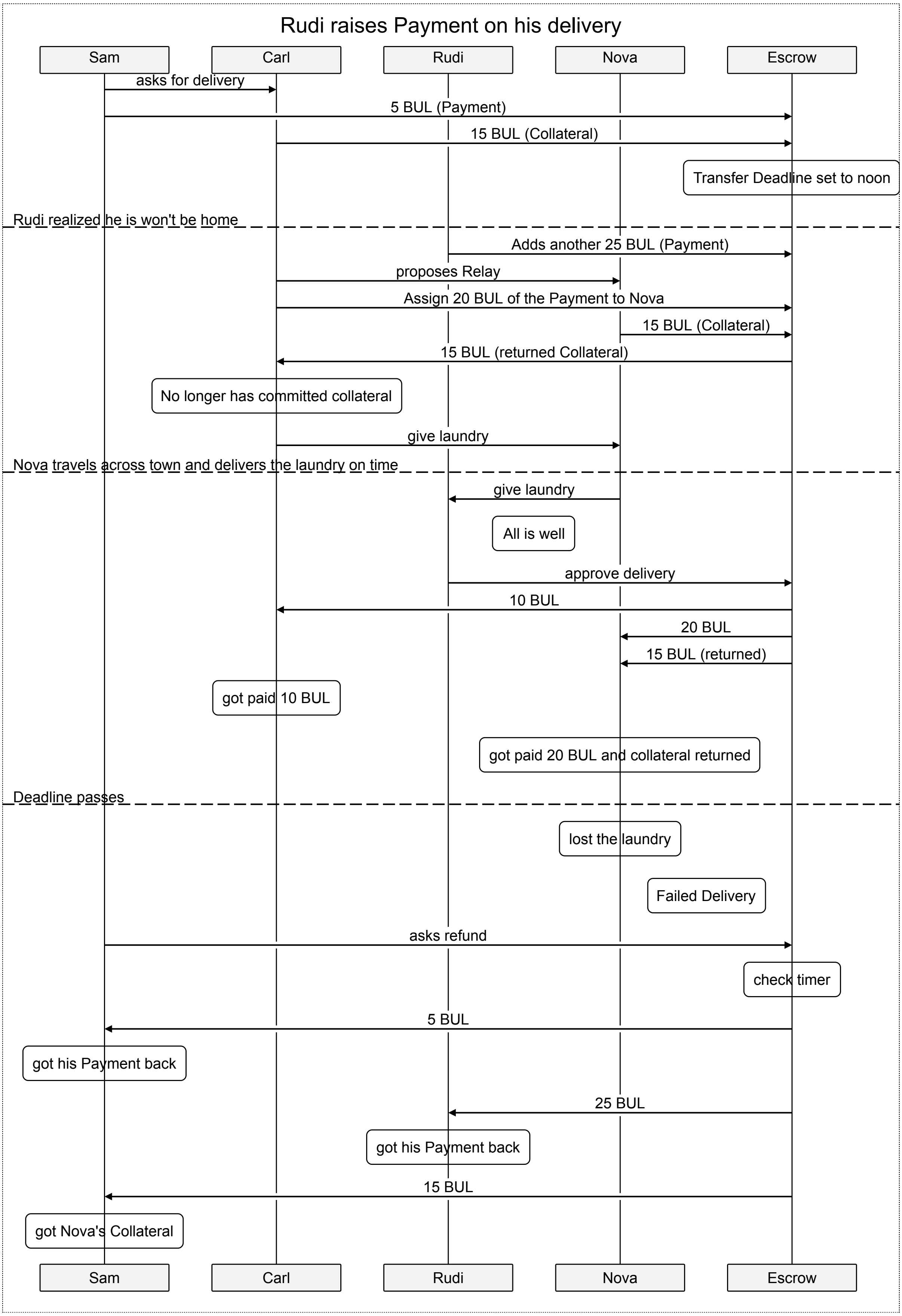
It is highly unlikely that a Courier will be willing to reach him on the other side of town for the originally offered 5 BULs, so Rudi raises the Payment. He adds another 25 BULs to the virtual escrow. Along with the 5 BULs that Sam committed earlier, they bring the total Payment to 30 BULs.

Let's imagine that this happens after Carl picks up the clothes, as he is sitting in the Cafe. He still meets Neo, but the only reason Neo was willing to make the delivery is that he lives right next to Rudi, so that's not an option. Carl could decide that the new increased Payment is worth his time and make the trip himself, but he is tired.

Fortunately, he has a friend who works in the same building as Rudi. Her name is Nova, and Carl is willing to carry the clothes all the way to the nearest train station to meet her just before she boards the train to work. For her help, Carl promises Nova 20 of the 30 BULs in the escrow. He also requires that she cover his Collateral of 15 BULs, which she is happy to do as she becomes the new Custodian.

Note that Rudi, who added the extra 25 BULs, is not our original Launcher. That's Sam. So if the delivery fails the virtual escrow will divide the promised Payment of 30 BULs and return 5 to Sam and 25 to Rudi. The Collateral, in this scenario, still goes to Sam.

Also note that if the delivery succeeds Carl is now paid 10 BULs instead of 3. He deserves them – not only for carrying the clothes all the way to the station but also for thinking of Nova. That was a good idea. Later we will see how L2 helps participants find and securely communicate with one another so that Senders, Recipients, and Couriers can easily find one another and cooperate even without prior personal acquaintance.



Raising Collateral

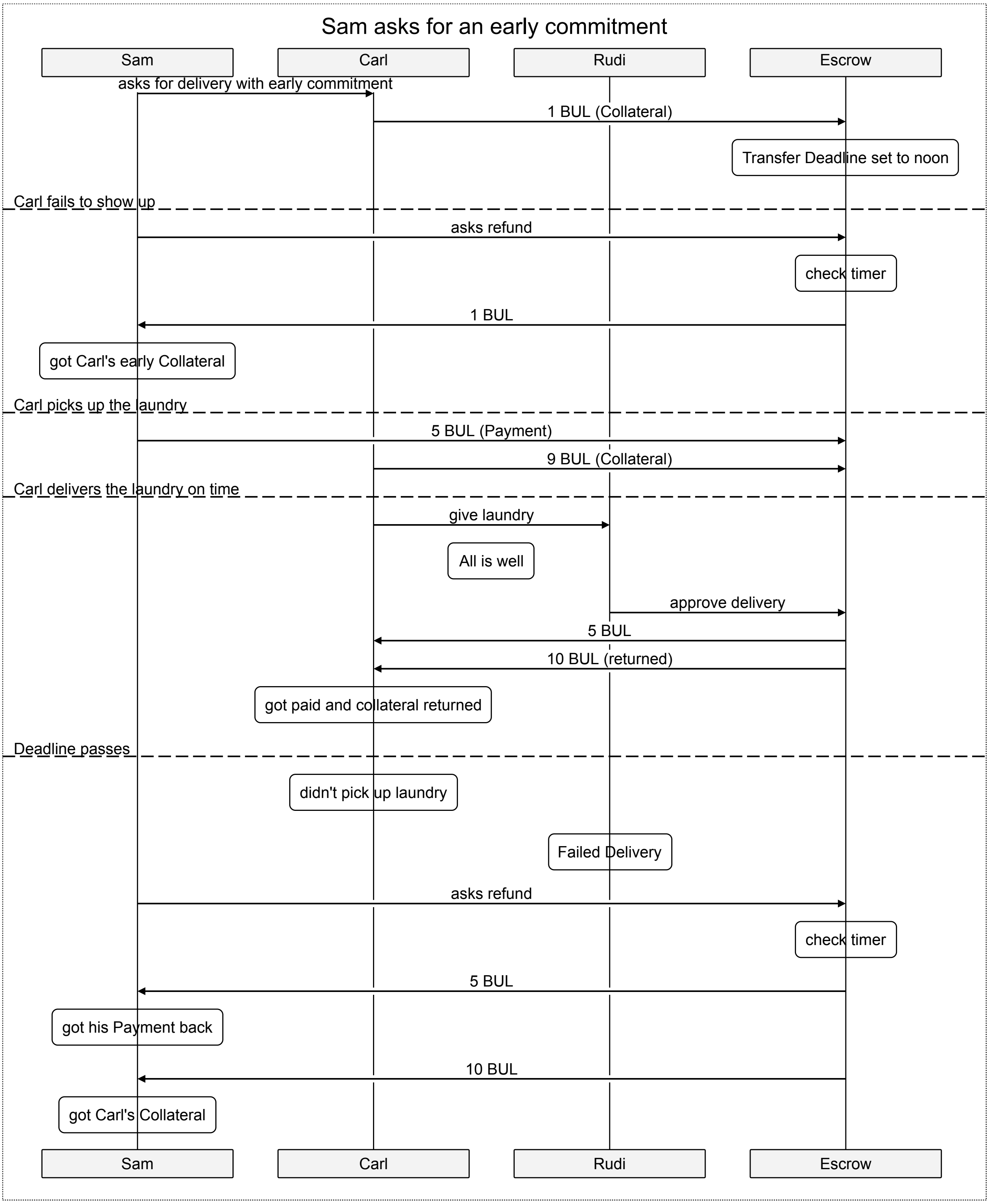
It is also possible to increase the amount of BULs deposited as Collateral. This is useful when either party requires some commitment in advance, as a way to ensure the other party is committed to the delivery even before it is picked up.

Example: Incremental Collateral On A Collect Delivery (Sender, Courier, Recipient)

Let's imagine that Sam doesn't trust Carl that much. Once the Paket changes hands this becomes inconsequential — both sides are financially committed to the success of the delivery. But what happens until Carl shows up to pick up the clothes? Sam doesn't want to wait for the last minute to find out that Carl stood him up.

The solution is simple. Sam asks Carl to commit a fraction of the Collateral to the virtual escrow, let's say 1 BUL. Now Carl has skin in the game. If he doesn't show up, he will lose his Collateral. When he does show up, he only needs to add 9 BULs of Collateral.

Note that if Carl commits to the delivery, even for a single BUL's worth, this makes him, by definition, a partial Custodian of the Paket. He shares this Custodianship with Sam, who actually holds the clothes. This is only one case of shared Custodianship.



Linking Pakets

It is sometimes favorable to connect different Pakets together, and even nest multiple Pakets inside another Paket. This opens the door to a multitude of cool tricks such as aggregated or consolidated deliveries and automatic return deliveries (in case the original delivery fails).

Another obvious usage for linking Pakets is offering Payment pending a different condition. For example, when you want the speed up the delivery. This process is as simple as Launching a new delivery because that's exactly what it is.

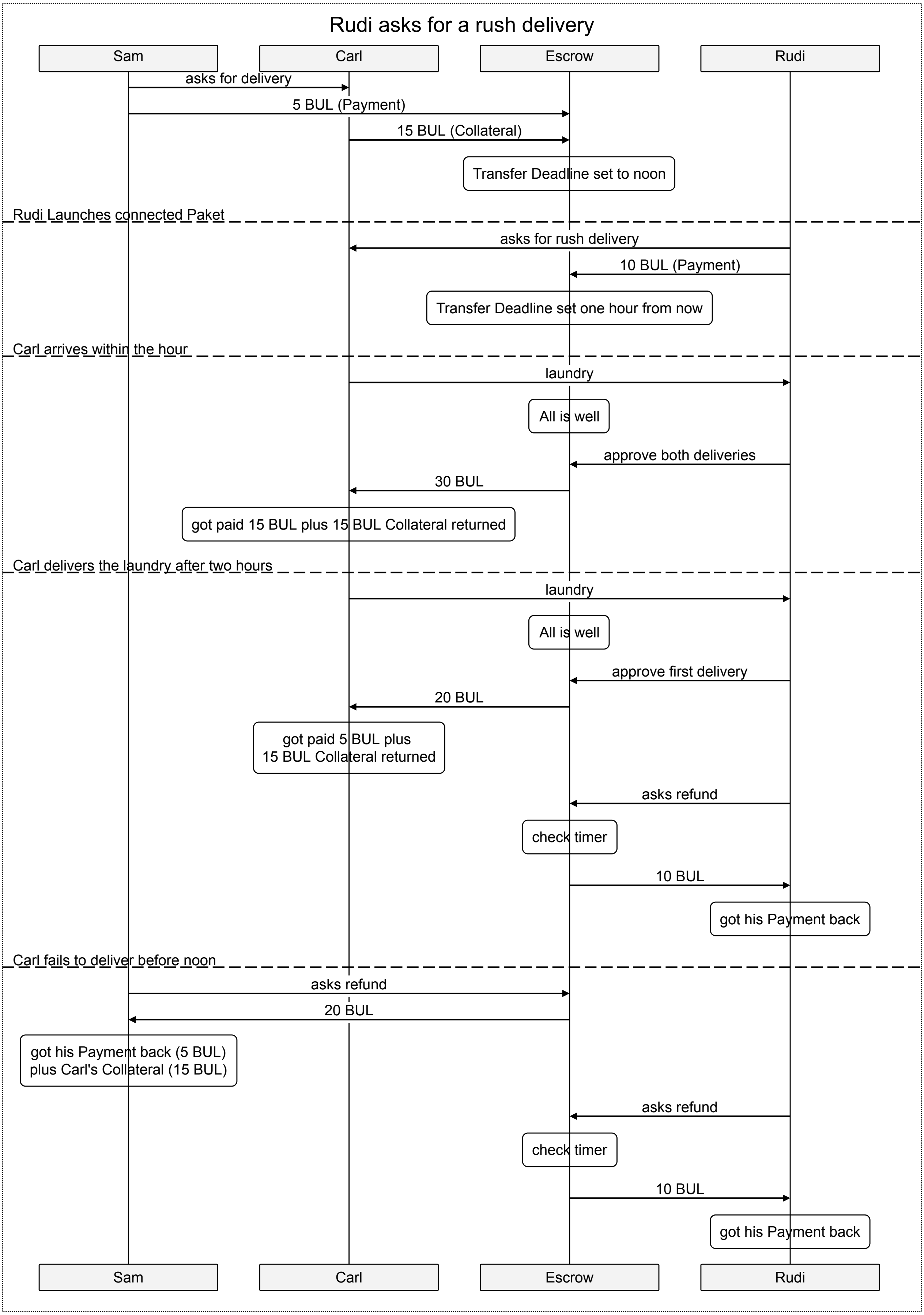
Example: Linked Paket - Simple Delivery Inside Collect Delivery (Sender, Courier, Recipient)

Let's imagine that Rudi doesn't have to go to his meeting right away. He has an extra hour. And he would really like to get his good jacket before leaving instead of waiting until the original Transfer Deadline.

No problem. He simply Launches a new simple delivery for a brand new Paket. Remember, Pakets are logical entities, and there is no problem with requesting the same single bag of clothes as two separate Pakets. In this case, Rudi promises the current Custodian of the original Paket, the one that currently has his jacket, 10 BULs if he delivers the clothes within the hour. When he commits those 10 BULs for rush delivery he knows that they will be added to the original 5 already committed by Sam. And let's say he doesn't demand any additional Collateral because he knows whoever has the clothes has already committed 15 BULs.

Carl, sitting at the cafe, sees this extra Payment and decides he is not that tired after all. For 15 BULs he is willing to put down his coffee and rush the clothes to Rudi's place himself.

Note that linked deliveries can succeed or fail independently. That's the whole idea - hinging the Payment and Collateral of the same deliverable on a different condition.


Rudi asks for a rush delivery
Sam
Carl
Escrow
Rudi
asks for delivery
5 BUL (Payment)
15 BUL (Collateral)
Transfer Deadline set to noon
Rudi Launches connected Paket
asks for rush delivery
10 BUL (Payment)
Transfer Deadline set one hour from now
Carl arrives within the hour
laundry
All is well
approve both deliveries
30 BUL
got paid 15 BUL plus 15 BUL Collateral returned
Carl delivers the laundry after two hours
laundry
All is well
approve first delivery
20 BUL
got paid 5 BUL plus 15 BUL Collateral returned
asks refund
check timer
10 BUL
got his Payment back
Carl fails to deliver before noon
asks refund
20 BUL
got his Payment back (5 BUL) plus Carl's Collateral (15 BUL)
asks refund
check timer
10 BUL
got his Payment back
Sam
Carl
Escrow
Rudi

Implementation

This layer is implemented in Solidity (https://en.wikipedia.org/wiki/Solidity) since our two leading L0 candidates use it for defining contracts.

Other Options

Based on the L0 platform we choose, there are several other possible implementations for this layer.

In any implementation, we need to consider the obvious trade-off is between readable verbosity, which might incur a severe cost (in gas) on the blockchain, and conciseness which might be cheaper but is less readable and harder to debug.

It is possible to implement L1 with a three-actor model (Launcher, Courier, and Recipient) instead of a two-actor model (Launcher and Courier).

In case of Relay, it is possible to release the Collaterals from all legs of the route only upon Proven Delivery. This exposes the Couriers to higher risk and has the side effect of decreasing velocity (more BULs remain “stuck in the pipes” for longer periods).

In case of Relay, it is possible to release Payment on every leg of the route, so that every Courier gets paid immediately upon completing his leg. This exposes the Couriers to lower risk but also lowers their level of engagement in the success of the delivery. This can also change the direction in which Payment flows (the last courier getting paid the full sum, keeping some for himself and moving the rest back down the chain, as opposed to each courier getting paid upon completing his leg and forwarding a chunk of it to the next courier).

Pure Bitcoin Implementation

If we have no smart contract platform but only what Bitcoin offers, this is still doable:

1. Launcher composes and shares (but does not sign) the following transactions:
   a. Commitment transaction (T1) which moves the Payment from himself (Launcher) to a multisig address controlled by himself and the receiving Courier, and the Collateral from the Courier to the same address.
   b. Refund transaction (T2) which is timelocked, and moves the Payment and the Collateral from the multisig address, back to the Launcher.
   c. Payment transaction (T3) which is locked with an external condition or secret (The external secret can be supplied by the target of the delivery, it can be a signature from some Oracle, and it can even be taken on goodwill alone, but it should be trusted by all parties to unlock upon (and only upon) a successful delivery.)(S1), moving the Payment and the Collateral from the multisig to the Courier.
2. Courier signs and shares T2 and T3.
3. Launcher signs and shares T1, T2 and T3.
4. Courier signs and publishes T1 to the blockchain.

Every transaction has an overhead in data transfer, data storage, and miner fees, so payment channels between nodes can and should be used to reduce the number of transactions actually broadcasted and mined.

Layer 2 — Routing

Layer 2 (L2) establishes a medium upon which potential Launchers and Couriers can publish their requirements and capacity, make and match offers, and ultimately discover the optimal route for a Package under predefined requirements.

Elements

- Messaging network - a (preferably decentralized) messaging and publishing network, probably a distributed hash table
- Launchers - seeking to have goods delivered in exchange for BULs
- Couriers - seeking to deliver goods in exchange for BULs

Description

Layer 2 describes the creation of a single Route (composed of any number of Relays) in which a package (which corresponds to a layer 1 Paket) can travel between Sender and Recipient and following a set of predefined conditions. Some of these conditions are a part of Layer 1 (Payment, Collateral and Delivery Deadline) and some are new, free-form conditions, which can be anything from delivery in refrigerated containers to KYC regulations.

Delivery Properties

Any offer made by either party, Launcher or Courier, specifies delivery properties such as location, pickup and delivery times, encumbrance and specific requirements (such as refrigerated containers). For a route to be valid, all properties must match and be accepted by all Couriers along it.

- L1 related properties
  - Payment
  - Collateral
- Time (source and destination): the time window in which a package can change hands at the specified location. Note that this is different from the Transfer Deadline, which is likely to be set right after the closing of the destination time window.
- Location (source and destination): can be accurate GPS coordinates or arbitrary descriptions (e.g., wherever Rudi hangs out these days).
- Encumbrance: can be specified in grams and liters but also arbitrarily (e.g., whatever fits in a manila envelope).
- Misc.: can be binary (must be delivered upright), ranged (must be kept between 5 and 7 degrees Celsius) or a list (countries through which the package must not pass).

While Payment and Collateral are very clearly defined (as a specific number of BULs), other properties are deliberately more fluid. We believe the protocol has to supply sufficient flexibility to allow for the organic growth of standards from within the community and in response to actual usage. Different user applications written above this layer will likely use predefined properties with enumerated and predefined sets of values, to ensure streamlined cooperation and interoperability.

Network Messages

There are two types of messages that can be broadcast to the Network, which for this purpose serves as a decentralized bulletin board.

- Courier Capacity - this is a general statement, made by a potential Courier, specifying his capacity and willingness to perform deliveries. This message includes ranges of when and where the Courier is willing to pick up packages, when and where he is willing to deliver them to, the expected Payment and offered Collateral, and any other property of

the delivery the Courier is capable of (from encumbrance limitations to refrigerated containers).
- Launch Request - this is a specific statement, made by a potential Launcher regarding a package. It specifies all the conditions for the delivery of that package; the when and where of pickup and delivery, the offered Payment and the required Collateral, and all the relevant properties of the package.

In addition, there are messages that can be sent privately between participants. This is simple, direct and discreet P2P messaging. All private messages are signed and therefore authenticated using the same keypair as the broadcast messages which is also the keypair used in layers 1 and 0, thus pegging each entity throughout the different layers.

Routing Methods

There are two different routing modes:

- Pre-Determined – the full path, including all participants, Payments, and Collaterals, is defined in advance.
- Opportunistic – the path and Payment are optimized in real-time while the package is en-route, which can potentially be much faster and cheaper

Once more, it is important to note that the user is not expected to use this layer directly and that the following “bare bones” examples are only provided as an explanatory device.

Example: Static Route (Sender, Courier, New Courier, Network, L1)

Let's imagine Sam, our Sender, doesn't have a working relationship with Couriers (like in the previous examples). Instead, he connects to the Network and starts reading Capacity messages made by Couriers. Maybe he finds a single Courier that meets his requirements:

- Corina, a Courier who is willing to move any bag from any location to any location in the neighbourhood for 5 BULs, on weekdays between 10 AM and 2 PM, as long as the content of the package is inspectable (will not deliver illegal or dangerous packages), doesn't weigh more than 5 kilogram, with a Collateral of up to 20 BULs.

It is also possible that Sam can find multiple Couriers whose Capacities can be combined, like so:

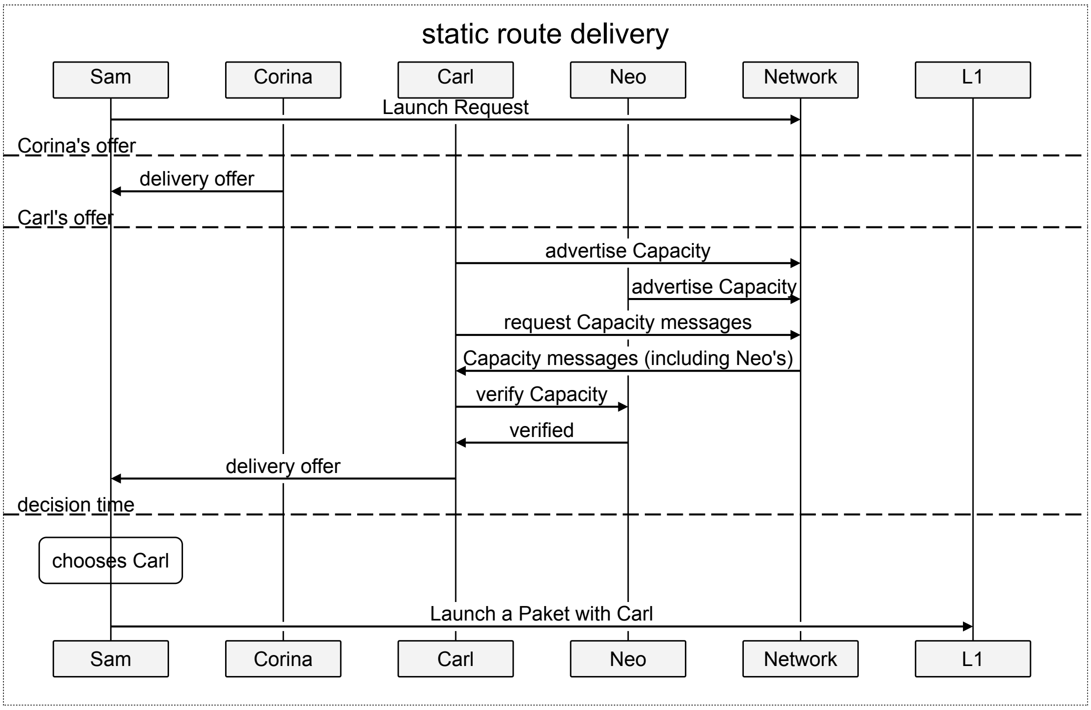
- Carl, willing to pick up clothes this morning and take them, within the hour, as far as the local cafe for 3 BULs with up to 50 BULs Collateral.
- Neo, willing to pick up any medium sized package from the area and deliver it to anyone who lives on his street when he comes back home for lunch, for a Payment of 2 BULs with up to 15 BULs Collateral.

But let's imagine that Sam does not find such a clever combination. Or maybe he doesn't want to be bothered with combinations. After all, he is just looking for a single Courier who is willing to commit to the delivery. So he publishes his own request for delivery on the Network. His message contains all the important details about the delivery: when and where it should be picked up from him, when and where it should be handed over to Rudi, how many BULs are offered as payment, how many are demanded as Collateral, how big and heavy the package is and how important it is for it not to be tossed around so as not to ruin Sam's legendary ironing work:

- Delivery from Sam's laundry shop (pickup anytime during working hours) to Rudi's home, until noon, for 5 BULs with 15 BULs Collateral. It's a bag with clothes, weighs about 2 kilograms, and the clothes are perfectly ironed and expected to arrive in the same condition.

Now it's up to the Couriers. So Corina can immediately offer to carry out this delivery, and Sam can close the deal with her. But let's inspect the slightly more complex scenario in which Carl wants to fulfil this request. Carl knows that the delivery is outside his Capacity because he only wants to go as far as the cafe, and, just like Sam, there are two things he can do about it: either look for Capacity messages (could be a single Courier or a combination of Couriers) or publish a new Launch Request.

Let's assume the first: Carl can see Neo's message, so he can talk to him and reach an agreement, making sure Neo can deliver the clothes without tossing the bag around. And if Carl doesn't trust Neo, maybe because he is a new Courier and has no history of deliveries (remember, pseudonymous immutable history is always inspectable on the blockchain), he can use L1 to demand a small Collateral. In any event, once an agreement is reached and Carl feels assured of his ability to complete the delivery with Neo's help, Carl can approach Sam and close a deal. Maybe he can even offer a better price than Corina.

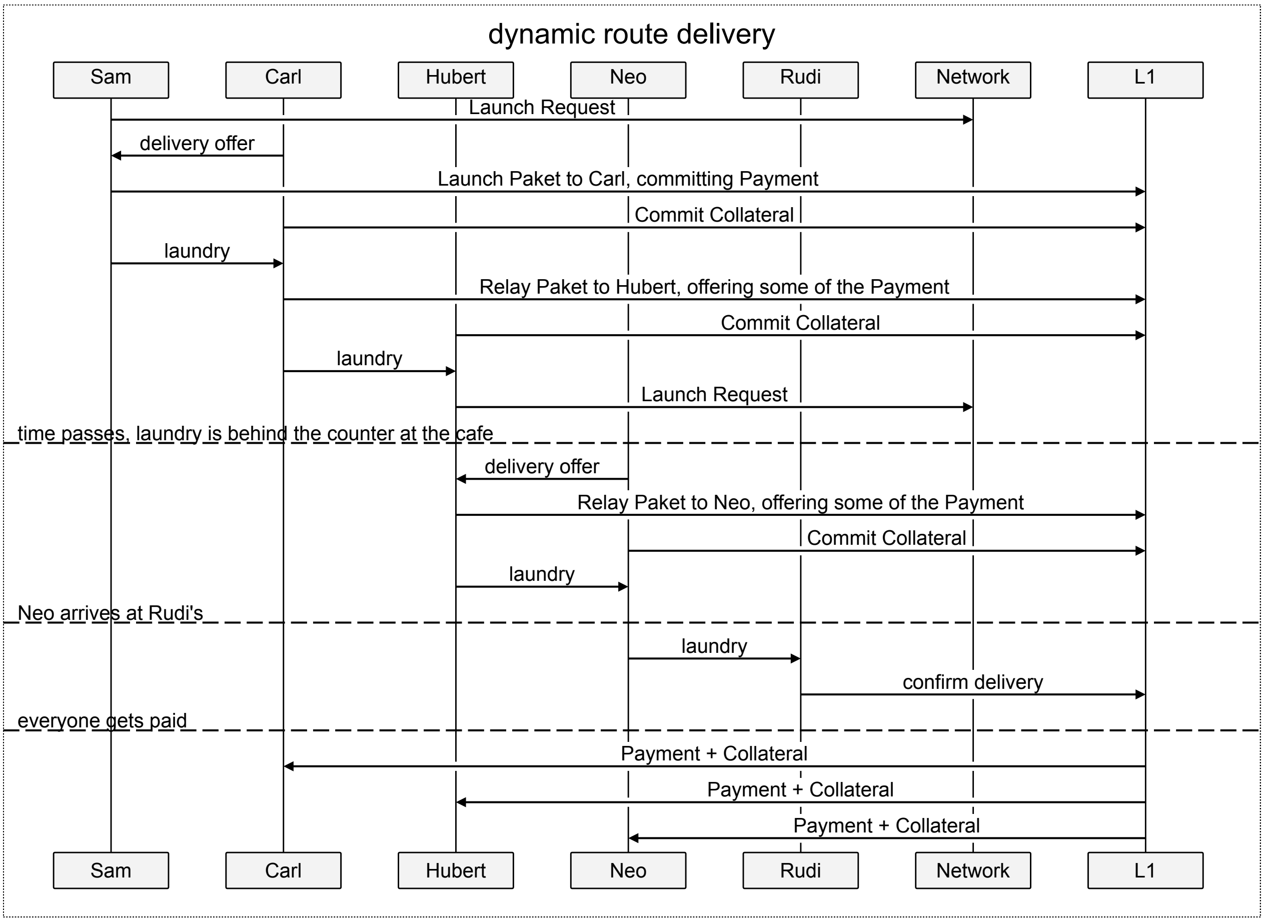


Example: Dynamic Route (Sender, Courier, Hub, New Courier, Network, L1)

In the previous example, Carl was not willing to commit to the delivery before he knew Neo was willing to commit as well, because he was uncertain of his Capacity to complete the delivery without him. But this is not always the case. If the neighbourhood has a good coverage of Couriers, Carl can be fairly certain of his ability to find one at a reasonable price.

The only problem is that Carl doesn't want to carry a bag of clothes with him until he finds a Courier. And this is where Hubert comes in. Remember that cafe Carl sits in? Hubert sits there all day. In fact, he owns the cafe. And while Hubert is unwilling to go anywhere else, he is willing, for the right Payment, to take the clothes from Carl and keep an eye on them until the new Courier picks them up. He is even willing to place a collateral because he knows his cafe is filled with potential Couriers. As a matter of fact, Rudi himself sometimes visits the Cafe - wouldn't it be nice if he could pick up his own clothes, earning a few BULs along the way.

So Carl can commit to the delivery without talking to anyone. He will pick up the clothes from Sam, deliver them to the Cafe where he Relays the package to Hubert, who will relay it to the next Courier (who may or may not be the final Courier who delivers the clothes to Rudi) once such a Courier is found.



Hubs

You probably noticed, in the example above, that Hubert is a special kind of Courier. He does not move the package in space, like an ordinary Courier, but instead moves it only in time. The Capacity to hold packages safely for a Payment (and against a deposited Collateral) adds a lot of flexibility to the system, mostly in offering Couriers the means to Relay packages to other Couriers without meeting them. Optimized paths are likely to organically form around well-functioning Hubs.

But the pressure for optimization goes both ways. Launch Requests, Courier Capacities, and actual deliveries (both failed and successful) are all observable and analyzable by anyone, anywhere, anytime. Their very existence paints a very clear picture of the market's demand for Hubs at specific locations. One can easily generate heat-maps, for example, that show the estimated profitability of just sitting in one place and keeping an eye on packages.

The utility of Hubs will be further explored in layers 3 and 4, but even at this layer, it is clear that Hubs are a driving force for matching, creating, and optimizing routes. In the example above, Hubert's cafe is such a Hub, and it is highly conceivable that it is Hubert himself who sees Sam's Launch Request and who coordinates between Carl and Neo.

Example: Hub Driven Route, Two Pakets Solution (Sender, Hub, Two Independent Couriers, Network, L1)

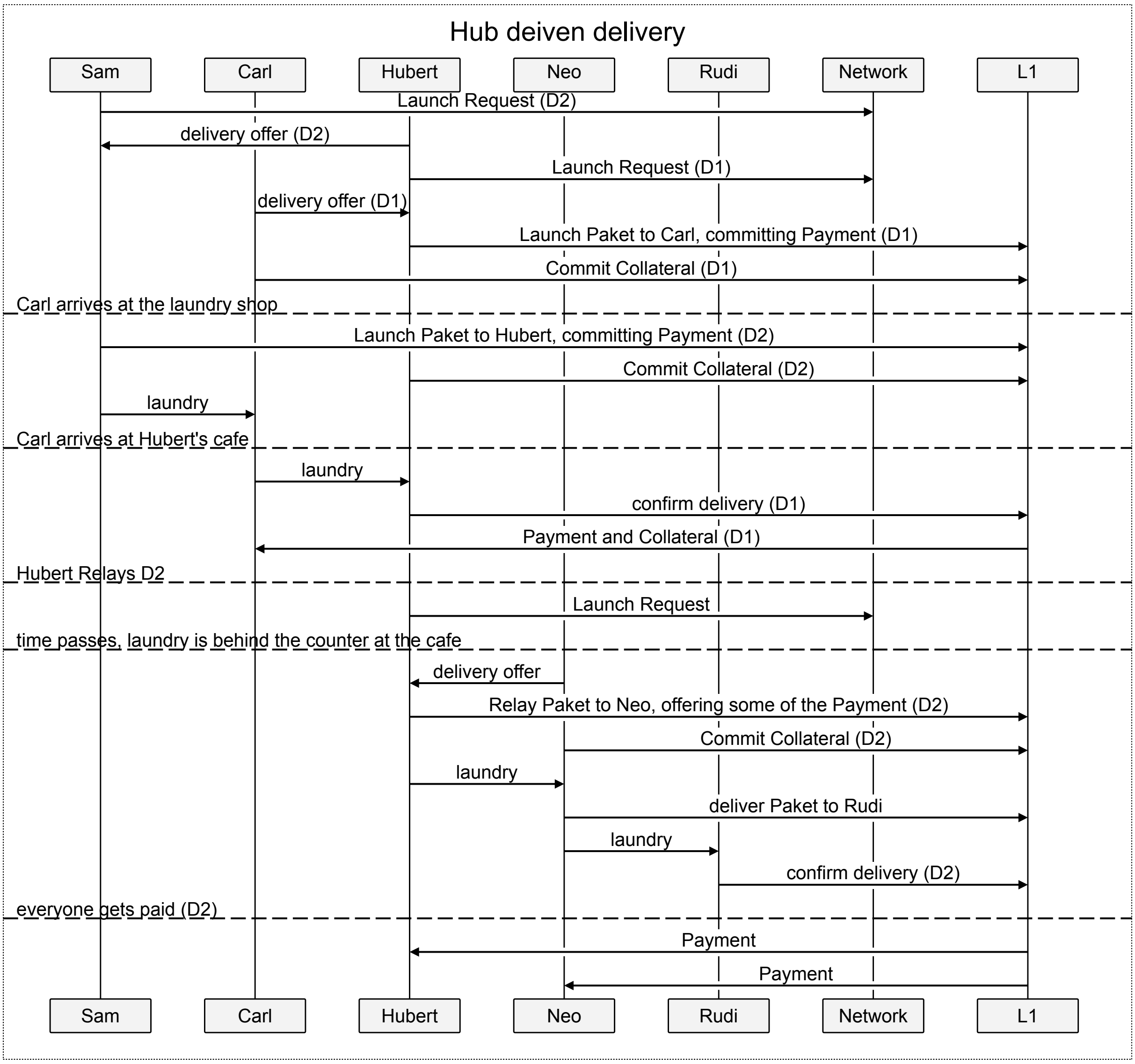
Let's imagine that Carl does not respond to Sam's Launch Request, but Hubert sees it and wishes to profit from it. He could look at Courier Capacities, discover Carl and ask him to bring him the package, but this has two important implications:

- Carl will receive his Payment only once the package is accepted by Rudi.
- The entire Payment will be promised to Carl, who can decide to complete it without Hubert.

So let's imagine Hubert decides to do things a bit differently. He gives Sam a delivery offer and then broadcasts a new Launch Request asking for the package to be delivered from Sam to him for 2 BULs. He may ask for the same Collateral Sam demands of him, but he can also make do with less. Or demand more. All in accordance with the risk he perceives. Let's imagine Sam doesn't see much risk and only asks for a 10 BULs Collateral.

Carl sees this Launch Request, which falls well within his Capacity, and decides to fulfill it, so Hubert and Carl finalize an L1 delivery (D1). The mindful reader will notice that when Carl arrives at the laundry shop and asks Sam for the clothes, Sam will demand 15 BULs of Collateral, while Carl is only committed (and is only willing to commit) 10. So Sam only gives Carl the clothes after he signs another L1 delivery **with Hubert** (D2), who commits the required 15 BULs, knowing that if Carl fails to bring him the clothes, he will only receive 10. That's the risk he is willing to take (probably because he does this a lot and most of the time it works). Note that at this stage, Carl is the Custodian of D1, but Hubert is the Custodian of D2. Carl is committed to Hubert, but Hubert is committed to Sam.

When Hubert confirms getting the clothes from Carl, the first L1 delivery, D1, is over and done. Carl gets both Hubert's Payment and his own Collateral. The second L1 delivery, D2, is still in the air. Hubert can now continue as in the previous example, with Hubert broadcasting a Launch Request and Neo answering it.



This example also shows how different risks can be applied to different legs of the route while keeping Collateral and Payment intact. The responsibility for the package and its delivery is clear at any and every point, with everybody always paying exactly what they agreed to pay and getting exactly what they paid for.

Implementation

We are currently looking at existing technologies, from Freenet (https://freenetproject.org/index.html) to twister (http://twister.net.co/), Matrix (https://matrix.org/), Whisper (https://github.com/ethereum/wiki/wiki/Whisper) and even IRC (https://en.wikipedia.org/wiki/Internet_Relay_Chat) for a reliable messaging infrastructure (see also A collection of peer-to-peer decentralized projects (https://github.com/moshest/p2p-index#communication)).

Security

Communication has to be secure and easy to authenticate. We use asymmetric cryptography (keypairs) to establish pseudonymous entities with observable and analyzable history and to encourage the organic growth of reputation. The same asymmetric cryptography keys are also used to link entities on this layer with entities in any other layer, with L1 being the most relevant in this case.

Protecting The Offer Publishing Platform layer 3 is responsible for handling this problem. There is no risk of actors reneging once they have signed the commitment transaction, because they are committed, and their personal history is secured on top of the blockchain from layer 0.

Spam is a potential problem which is easily solved by demanding some small Proof of Work, probably in proportion to the offered or requested payment (see hashcash (https://en.wikipedia.org/wiki/hashcash)), but consideration must also be given to mobile SPV clients.

Layer 3 – User

Layer 3 (L3) deals with the human usage of the framework, as it consolidates tools and applications that are used to interact with lower layers. Participants in the network are expected to use Layer 3 applications almost exclusively.

Scope

Layer 3 is responsible for providing network participants with easy to use tools that allow them to:

- Send a Package to anyone in the world, including Recipients that are not aware of the PaKeT network (be a Sender)
- Order a package and have it delivered (be a Launching Recipient)
- Publish your delivery capabilities, find packages to deliver and transport them between other parties (be a Courier)
- Create and manage a Hub
- Gather and analyze information from the network, such as demand for Couriers or prices in specific areas

Development will start with minimally viable applications that allow early users to make immediate use of the network; namely sending and receiving packages, and delivering them in exchange for BULs.

While we will continue to extend and improve our applications over time, adding value to them and the network as a whole, our primary focus is on supporting the community in creating its own tools and applications according to actual needs. Eventually, we intend to stop developing applications altogether (see Organic Death) and rely solely on the community and the applications it develops and maintains.

Simplifying Usage

L3 ensures that the use of the system is simple and immediate. Layers 1 and 2 are decidedly simple, but they do allow an extremely wide range of possibilities, which can be overbearing for end users. L3 applications, at least the early ones, should strive to hide most of this flexibility to provide basic and simple capabilities that satisfy basic and simple user needs.

Description

We are developing several different types of applications, and are expecting more types to be added in the future.

Wallet Application

Allows users to hold and transfer BULs.

Recipient Application

Allows users to specify their delivery requirements, choose from received delivery offers, identify their selected Courier upon arrival, Launch the package, track it in transit and be notified of its success or failure (with automatic refund). Must include an embedded wallet.

The application will have an option to offer an initially low price and raise it over time.

It will also make extensive use of camera capabilities (mostly on mobile devices) for identification of Couriers and Packages as well as for documenting and proving transport and delivery.

Sender Application

The only fundamental difference between the Recipient and the Sender application is that the latter makes only Collect Deliveries.

It will also have a return-to-sender option, which uses BULs from the Collateral to fund a return delivery.

A very nice addition would be the use of RFID-NFC to create physical tags (e.g. stickers) that can hold a key that enables automatic confirmation and acceptance at every location with simple NFC readers (e.g. smartphones). Take a look at: http://www.nxp.com/docs/en/brochure/make-your-products-smarter-with-RFID-NFC.pdf (http://www.nxp.com/docs/en/brochure/make-your-products-smarter-with-RFID-NFC.pdf)

Courier Application

Allows users to publish their capacities as Couriers, view requested deliveries and perform them. Must includes an embedded wallet.

Hub Application

Allows users to start and manage a Hub.

Data Presentation Application

Allows users to collect, analyze and view statistics about the network:

- Suggested prices for both supply and demand sides
- Network statistics and growth
- Maps of global network coverage
- Heat map of transfer utilization and delivery prices

Implementation

It's rather early to decide, but our current preference is for applications written in Python for its wide adoption, cross platform support, and because it does a very good job at encouraging simple and readable code.

Layer 4 – Organization

Layer 4 (L4) deals with the different organizations that interact with the PaKeT network.

Scope

Layer 4 describes organizations that use lower layers, especially but not exclusively L3 applications, to offer and perform relevant services and improve the PaKeT ecosystem.

Description

The lower layers provide ample possibilities for the creation of organizations that build important services on top of them, and many classes of organizations can blossom in this ecosystem. Following is a description of several of the more obvious possibilities.

Hub Organization

Hubs give the network tremendous value. They are a both the "glue" that links Couriers with each other, and at the same time, they act as a "lubricant" that reduces friction when transferring the packages. They allow two Couriers, for a relatively small fee, to exchange packages between them without the need to coordinate a specific time to meet, as one can leave the package in the secure Hub for the other to pick up at a later time.

Hubs can be created by a single operator, by a group of people, or by a company. A local grocery store owner can become a Hub as an added source of income, while the entrepreneurial owner of a delivery company can decide to add Hub, or even several, as additional service he offers and as a tool to improve his operations.

Delivery Company

Existing delivery companies can interface with the system in two main ways: they can be used by other Couriers to assist in deliveries, even without their knowledge, and they can offer their services over the Network for BULs. It is also very likely that entirely new delivery companies will form within the ecosystem in response to increased demand.

Courier Group

Imagine a city with several Couriers. All of them are the supply side in this city delivery market. They all compete over providing a service of deliveries, earning BULs as payment and committing BULs as Collateral. Imagine that five of them, who use bicycles to make deliveries, decide to cooperate. They form a small group (or a co-op, or a union) and immediately benefit from:

- Decreased chances of failed deliveries – if one of them suffers a flat tire he can ask one of his friends for help
- Larger Collateral pool – they can pool their BULs and compete over higher Collateral deliveries
- Synchronization – by dividing the city between them they can increase utility and reduce unneeded competition
- Reputation – as this increased utility is publicly visible on the blockchain, they can become more attractive to Launchers, who will not only prefer them over the competition but may also be willing to offer higher Payment and require lower Collateral

Insurance Company

Insurance companies, both existing and new, can use the L1 concept of Collateral to insure Launchers, Senders, Recipients, and Couriers.

They can offer Couriers the option to pay their Collateral for a well-calculated premium, having full access to all the delivery related information on the blockchain, allowing Couriers to compete over high Collateral deliveries or to deliver multiple packages at the same time.

Foundation

For the bootstrapping phase, we intend to form a non-profit organization that will initially hold most of the minted BULs and use them to promote and incentivize the budding network. The activities of the foundation may include:

- Operating a non-profit, subsidized Hub
- Operating a faucet that gives BULs for soft commitments of Courier capacity
- Offer bug and feature bounties to promote the technological aspects of the system

Implementation

Software to support organizations is likely to be Web-based. Due to the exceptionally high technological turnover rate in this field, we will postpone choosing a framework for as long as we can.

Legal Disclaimer

This preliminary paper is for information purposes only and may be subject to change. We cannot guarantee the accuracy of the statements made or conclusions reached in this preliminary paper and we expressly disclaim all representations and warranties (whether express or implied by statute or otherwise) whatsoever, including but not limited to:

- any representations or warranties relating to merchantability, fitness for a particular purpose, suitability, title or non-infringement;
- that the contents of this document are accurate and free from any errors; and
- that such contents do not infringe any third party rights.

We shall have no liability for losses or damages (whether direct, indirect, consequential or any other kind of loss or damage) arising out of the use, reference to or reliance on the contents of this preliminary paper, even if advised of the possibility of damages arising.

This preliminary paper may contain references to third party data and industry publications. As far as we are aware, the information reproduced in this preliminary paper is accurate and that the estimates and assumptions contained herein are reasonable. However, we offer no assurances as to the accuracy or completeness of this data. Although information and data reproduced in this preliminary paper are believed to have been obtained from reliable sources, we have not independently verified any of the information or data from third party sources referred to in this preliminary paper or ascertained the underlying assumptions relied upon by such sources.

As of the date of publication of this preliminary paper, BUL tokens have no known or intended future use (other than on the PaKeT Project platform which is still under development).

No promises of future performance or value are or will be made with respect to BUL tokens, including no promise of inherent value, no promise of any payments, and no guarantee that BUL tokens will hold any particular value. Unless prospective participants fully understand and accept the nature of BUL’s business and the potential risks associated with the acquisition, storage and transfer of ERC-20 tokens such as BUL tokens, they should not participate in the token sale.

BUL tokens are not being structured or sold as securities. BUL tokens hold no rights and confer no interests in the equity of The PaKeT Project. BUL tokens are sold with an intended future use on The PaKeT Project’s platform and all proceeds received during the token sale may be spent freely by The PaKeT Project on the development of its business and the underlying technological infrastructure.

This preliminary paper does not constitute a prospectus or disclosure document and is not an offer to sell, nor the solicitation of any offer to buy any investment or financial instrument in any jurisdiction. BUL tokens should not be acquired for speculative or investment purposes with the expectation of making an investment return.

No regulatory authority has examined or approved any of the information set out in this preliminary paper. No such action has or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of this preliminary paper does not imply that applicable laws or regulatory requirements have been complied with.

Participation in the token sale carries substantial risk and may involve special risks that could lead to a loss of all or a substantial portion of your contribution. Further information about the risks of participating in the token sale is set out in the Future Tokens Agreement to which you are a party or in the Token Sale T&Cs. Please ensure that you have read, understood and are prepared to accept the risks of participating in the token sale before sending a contribution to us.

The token sale and/or BUL tokens could be impacted by regulatory action, including potential restrictions on the ownership, use, or possession of such tokens. Regulators or other competent authorities may demand that we revise the mechanics of the token sale and/or the functionality of BUL tokens in order to comply with regulatory requirements or other governmental or business obligations. Nevertheless, we believe we are taking commercially reasonable steps to ensure that the token sale mechanics and issue of BUL tokens do not violate applicable laws and regulations.

CAUTION REGARDING FORWARD-LOOKING STATEMENTS

This preliminary paper contains forward-looking statements or information (collectively “forward-looking statements”) that relate to our current expectations of future events. In some cases, these forward-looking statements can be identified by words or phrases such as “may”, “will”, “expect”, “anticipate”, “aim”, “estimate”, “intend”, “plan”, “seek”, “believe”, “potential”, “continue”, “is/are likely to” or the negative of these terms, or other similar expressions intended to identify forward-looking statements. We have based these forward-looking statements on current projections about future events and financial trends that we believe are relevant to our financial condition, results of operations, business strategy, financial needs, or the results of the token sale.

In addition to statements relating to the matters set out here, this preliminary paper contains forward-looking statements related to The PaKeT Project’s proposed operating model. The model speaks to our objectives only, and is not a forecast, projection or prediction of future results of operations.

Forward-looking statements are based on certain assumptions and analysis made by The PaKeT Project in light of its experience and perception of historical trends, current conditions and expected future developments and other factors it believes are appropriate, and are subject to risks and uncertainties. Although the forward-looking statements contained in this preliminary paper are based upon what we believe are reasonable assumptions, there are risks, uncertainties, assumptions, and other factors which could cause our actual results, performances, achievements and/or experiences to differ materially from the expectations expressed, implied, or perceived in forward-looking statements. Given such risks, prospective participants in the token sale should not place undue reliance on these forward-looking statements.

WARNING: You should *not* invest in any Token Sale unless you are prepared to sustain a total loss of the money you have invested plus any commission or other transaction charges.

DISCLAIMER: This website and the team behind it do not provide any investment advice. This material has been prepared for informational purposes only, and is not intended to provide, and should not be relied on for, investment, tax, legal or any other advice. You should consult your investment, tax, legal and accounting advisors before engaging in any investment.