Product Overview

A technical overview of xCurrent

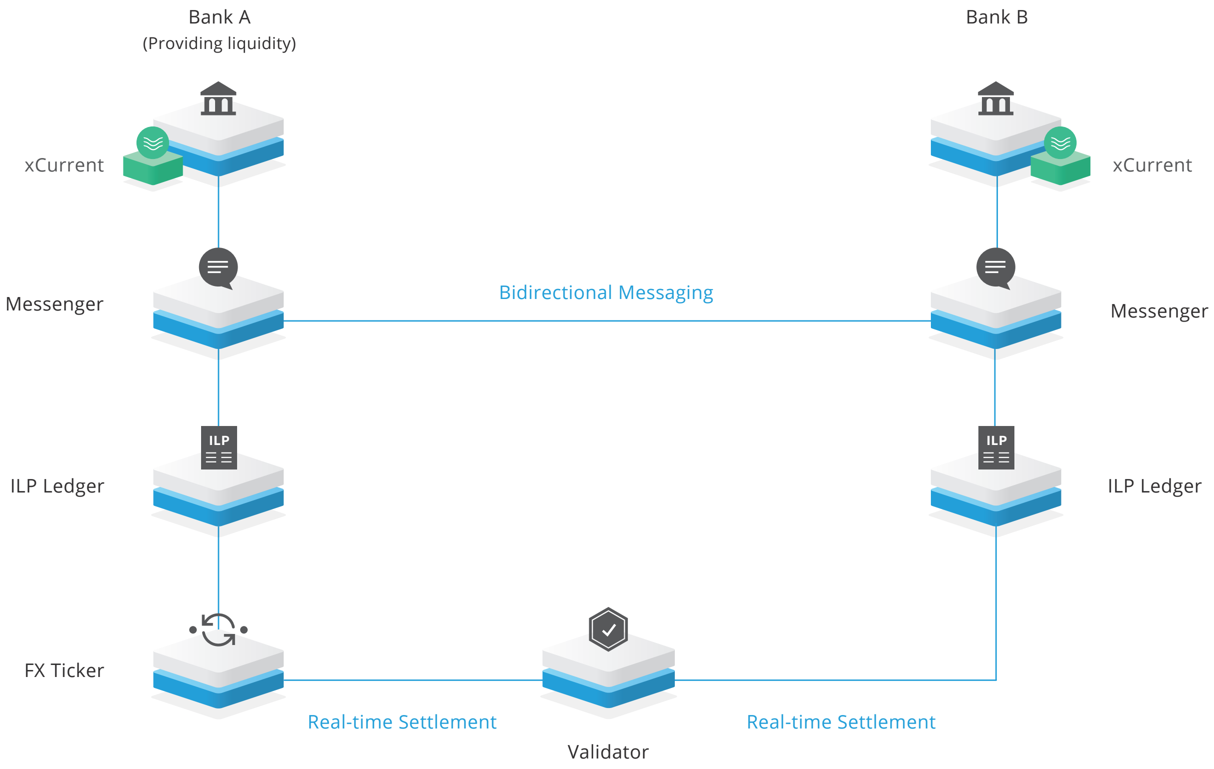
October 2017

One frictionless experience to send money globally

A consistent technology across a global network of financial institutions

Product Overview

Ripple's solution for banks, xCurrent, is built around an open, neutral protocol, Interledger Protocol (ILP), which enables interoperation between different ledgers and networks. It offers a cryptographically secure, end-to-end payment flow with transaction immutability and information redundancy. It is designed to comply with a bank's risk, privacy and compliance requirements. It is architected to fit within a bank's existing infrastructure, resulting in minimal integration overhead and business disruption.



There are four components of xCurrent:

MESSENGER

Messenger is the API-based, bidirectional messaging component of xCurrent. It connects to the beneficiary bank's Messenger instance to exchange KYC and risk information, fees, FX rates (if applicable), payment details and expected time of funds delivery. It packages this information and presents the entire cost structure to the originating bank, providing unprecedented visibility into the total cost of the transaction. If information is incorrect or missing, transacting parties will find out before initiating the transaction, drastically increasing STP rates. Additionally, banks can set fees and the FX rate for payments made with Messenger. FX rates are set in FX Ticker and queried by Messenger during the quoting process.

Each end-to-end payment has a payment ID that can be used to query the status of the payment at any point during payment execution, including funds settlement or thereafter – allowing for more effective troubleshooting for failed or delayed payments. The payment data exchanged through Messenger can be used to meet jurisdiction-specific regulatory needs and other enhanced services. Messenger uses Transport Layer Security (TLS) v1.2 for secure communication with existing bank systems, partner Messenger instances and ILP components of xCurrent.

VALIDATOR

Validator is a component of xCurrent that cryptographically confirms the success or failure of a payment. It coordinates the funds movement across ILP Ledgers of transacting parties in a way that removes all settlement risk and minimizes delays in settlement. Validator provides the single source of truth for the transacting counterparties while preserving the privacy of banking customers' identifiable payment information. Banks have the option of running their own Validator, using it for all their transactions, or relying on a Validator run by the transacting counterparty.

ILP LEDGER

ILP Ledger is a subledger of each transacting bank's general ledger. This component of xCurrent is utilized to the track the credits, debits and liquidity across the transacting parties. ILP Ledger enables transacting parties to settle funds atomically, which means the entire transaction settles instantly or not at all – no matter how many parties are involved.

ILP Ledger enables funds settlement in milliseconds. Further, the settlement risk is eliminated because the payment processes entirely or fails upfront. ILP Ledger is designed to provide transacting banks with 24/7, on-demand availability. The combination of these capabilities allows banks to profitably offer low-value, on-demand international payments products and services.

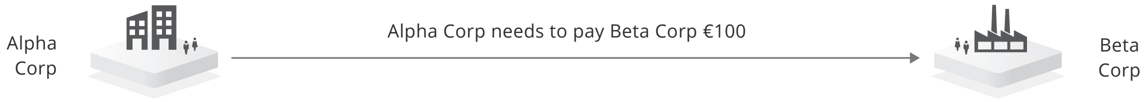
FX TICKER

FX Ticker is the component of xCurrent that facilitates the exchange between ILP Ledgers by enabling liquidity providers to post FX rates. This component provides the exchange rate between any pair of ledgers with which it is configured. Additionally, it keeps track of the account, currency and authentication credentials for each configured ILP Ledger. During the transaction, it coordinates transfers on ILP Ledgers for settlement, ensures the validity of an FX quote and transfers the payment amount to the beneficiary bank's ILP Ledger.

How It Works

Flow Of Funds

This section will use an example payment to demonstrate the flow of funds through xCurrent. In this example, Alpha Corp (in the U.S.) needs to pay Beta Corp (in the eurozone) a total of €100. Alpha Corp has an account with Dollar Bank in the U.S. and Beta Corp has an account with Euro Bank in the E.U. Both banks have integrated Ripple's software, xCurrent.



SETUP

To enable cross-currency flows via xCurrent, banks can leverage their existing nostro/vostro relationships with other banks and provide liquidity through their FX trading desks, or use external market makers to provide FX liquidity for exotic currency corridors. This example will refer to that function as the liquidity provider, whether it is the bank's FX organization or an external market maker.

As part of the set-up process, the liquidity provider ensures that the beneficiary bank account is prefunded with the local currency.

Dollar Bank's Ledger			
Account	Debit	Credit	Balance
Originator			$10,000
Liquidity Provider			
Fees			
Ripple Segregated Account			

Euro Bank's Ledger			
Account	Debit	Credit	Balance
Beneficiary			€3,000
Liquidity Provider		€200,000	€200,000
Fees			
Ripple Segregated Account			

Dollar Bank's Ripple ILP Ledger			
Account	Debit	Credit	Balance
Hold			
Liquidity Provider			

Euro Bank's Ripple ILP Ledger			
Account	Debit	Credit	Balance
Hold			
Liquidity Provider			

Each bank sets up a segregated account, the balance of which is reflected on ILP Ledger (a subledger to track the state of the liquidity provider's funds). Any contribution that a liquidity provider makes to the segregated account is reflected in the liquidity provider's account on ILP Ledger.

In this case, the liquidity provider makes €40,000 available for funding payouts of Ripple transactions. For a two-way flow, the liquidity provider would also prefund their dollar account and transfer some of the liquidity to the segregated account. For this example, only a payment going from Dollar Bank to Euro Bank is being shown.

Dollar Bank's Ledger

Account	Debit	Credit	Balance
Originator			$10,000
Liquidity Provider			
Fees			
Ripple Segregated Account			

Euro Bank's Ledger

Account	Debit	Credit	Balance
Beneficiary			€3,000
Liquidity Provider		€200,000	€200,000
	€40,000		€160,000
Fees			
Ripple Segregated Account		€40,000	€40,000

Dollar Bank's Ripple ILP Ledger

Account	Debit	Credit	Balance
Hold			
Liquidity Provider			

Euro Bank's Ripple ILP Ledger

Account	Debit	Credit	Balance
Hold			
Liquidity Provider		€40,000	€40,000

Once the segregated accounts are funded, the liquidity provider posts an FX quote to the originating bank. In this case, the offer is EUR/USD at 1.1429.

PAYMENT FLOW: INTEGRATED MESSAGING AND SETTLEMENT

When a payment is initiated by Alpha Corp, Messenger instances at both banks exchange information about Alpha Corp and Beta Corp for KYC/AML checks and sanctions screening. These CIP/PII fields are entirely configurable by both banks. The sending Messenger also queries Euro Bank for its processing fees for posting the payment for Beta Corp. It also obtains the exchange rate from the liquidity provider (could be the bank's FX desk).

Dollar Bank's Messenger compiles this information, adds their processing fees and presents the bank with the all-in cost of the transaction. Assuming that Dollar Bank's fees are $5, Euro Bank's fees are €5 and the EUR/USD rate at 1.1429, the total cost of sending €100 to Beta Corp would be $125. Once Alpha Corp accepts the charge, the payment is initiated. Dollar Bank debits Alpha Corp's account to the amount of $125, collects the $5 fee and credits the segregated account $120.

These funds are not yet credited to the liquidity provider. They are put on hold until Euro Bank provides proof to Validator that it has also put funds on hold that can be posted to Beta Corp.

Dollar Bank's Ledger

Account	Debit	Credit	Balance
Originator			$10,000
	$125		$9,875
Liquidity Provider			
Fees		$5	$5
Ripple Segregated Account		$120	$120

Euro Bank's Ledger

Account	Debit	Credit	Balance
Beneficiary			€3,000
Liquidity Provider		€200,000	€200,000
	€40,000		€160,000
Fees			
Ripple Segregated Account		€40,000	€40,000

Dollar Bank's Ripple ILP Ledger

Account	Debit	Credit	Balance
Hold		$120	$120
Liquidity Provider			

Euro Bank's Ripple ILP Ledger

Account	Debit	Credit	Balance
Hold			
Liquidity Provider		€40,000	€40,000

Euro Bank also puts €105 on hold and provides a cryptographic receipt to Validator proving that the condition has been fulfilled. This triggers Validator to instruct Dollar Bank to put the funds on hold from Alpha Corp and provide a cryptographic receipt of the hold. These receipts contain the cryptographic proof of the hold of funds, but do not contain any information about the banks, transacting parties or payment details.

Dollar Bank’s Ledger

Account	Debit	Credit	Balance
Originator			$10,000
	$125		$9,875
Liquidity Provider			
Fees		$5	$5
Ripple Segregated Account		$120	$120

Euro Bank’s Ledger

Account	Debit	Credit	Balance
Beneficiary			€3,000
Liquidity Provider		€200,000	€200,000
	€40,000		€160,000
Fees			
Ripple Segregated Account		€40,000	€40,000

Dollar Bank’s Ripple ILP Ledger

Account	Debit	Credit	Balance
Hold		$120	$120
Liquidity Provider			

Euro Bank’s Ripple ILP Ledger

Account	Debit	Credit	Balance
Hold		€105	€105
Liquidity Provider		€40,000	€40,000
	€105		€39,895

Once Validator receives proof that both banks have funds on hold, it triggers the settlement of funds – instructing both ledgers to release the holds and transfer the funds. This is an atomic process, meaning that both intra-bank settlement legs of the transaction happen simultaneously, to eliminate the settlement-leg risk.

Note on third-party validators: Theoretically, third parties could host a network of Validators reaching consensus through a byzantine-fault-tolerant (BFT) consensus algorithm without the risk of delivering sensitive protected data to a third party or publishing it to a public distributed ledger. The transaction details remain private to only the transacting banks, while Validator is used to verify whether certain crypto-conditions have been fulfilled (i.e., whether the funds are available for delivery).

Dollar Bank's Ledger

Account	Debit	Credit	Balance
Originator			$10,000
	$125		$9,875
Liquidity Provider			
Fees		$5	$5
Ripple Segregated Account		$120	$120

Euro Bank's Ledger

Account	Debit	Credit	Balance
Beneficiary			€3,000
		€100	€3,100
Liquidity Provider		€200,000	€200,000
	€40,000		€160,000
Fees		€5	€5
Ripple Segregated Account		€40,000	€40,000
	€105		€39,895

Dollar Bank's Ripple ILP Ledger

Account	Debit	Credit	Balance
Hold		$120	$120
	$120		$0
Liquidity Provider		$120	$120

Euro Bank's Ripple ILP Ledger

Account	Debit	Credit	Balance
Hold		€105	€105
	€105		€0
Liquidity Provider		€40,000	€40,000
	€105		€39,895

Once the transaction settles on both ILP Ledgers, Euro Bank collects the €5 fee and posts €100 to Beta Corp's account.

Once the funds post to Beta Corp's account, Dollar Bank is notified and can now provide a payment confirmation to Alpha Corp.

API Process Flow

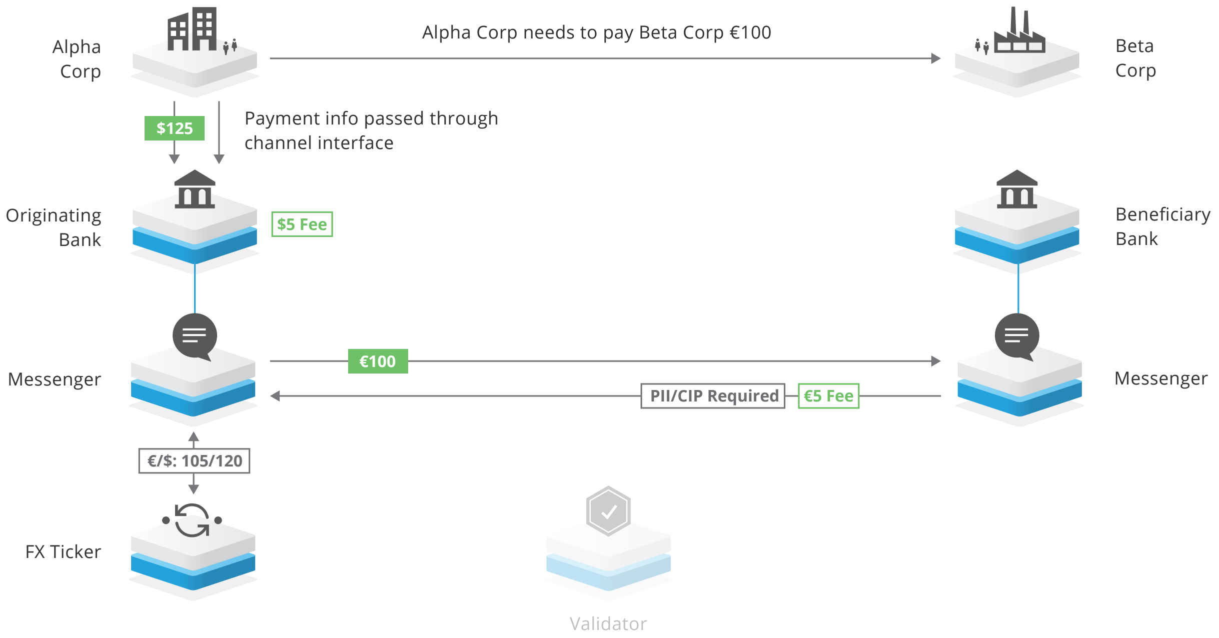
Executing the payment described in the Flow Of Funds section involves the following API requests to Messenger.

QUOTING PROCESS

1. The originator of the payment starts the process by providing information about the payment through an interface in the bank's client application (which could be part of an existing online banking system). At a minimum, this information must include the following:
   a. **Sender:** The originator of the payment.
   b. **Receiver:** The beneficiary of the payment.
   c. The **amount** and **currency** of the payment, and whether this is a "sender" or "receiver" amount:
      - **Sender Amount:** The specified amount is debited from the sender's account. Messenger calculates the fees and FX cost and debits them from the sender's account. The receiver's account is credited with the remaining amount.

- **Receiver Amount:** The receiver's account is credited with the specified amount. The amount is debited from the sender's account in the sender's currency. Messenger calculates the fees and FX cost – adding them to the amount debited from the sender's account. The originating bank uses the information provided by the originator to make a `Get Quote` request to its own Messenger instance.

2. The originating bank's Messenger instance makes a `Get Quote` request to the Messenger instance at the beneficiary bank to receive its portion of the payment, which includes its fees (€5 in the example above) and empty fields that indicate any additional information the beneficiary bank requires to process the payment.
3. Messenger instance at the originating bank gets the FX rate posted by the liquidity provider from FX Ticker.
4. The originating bank builds its portion of the payment, which includes its fees ($5 in the example above).



5. The originating bank receives a response to its `Get Quote` request and passes the quote to the initial sender to determine if the terms of the payment (which include the beneficiary and originating banks' fees and the FX rate) are acceptable. If the terms are acceptable, the originating bank makes an `Accept Quote` request. If the configuration of the beneficiary bank's Messenger has requested additional information about the payment, the originating bank provides that information in the `Accept Quote` request. (Additional payment information is not technically required, but for regulatory reasons institutions often require information similar to fields in pacs.008 or MT 103 messages to process payments.) Messenger generates a payment

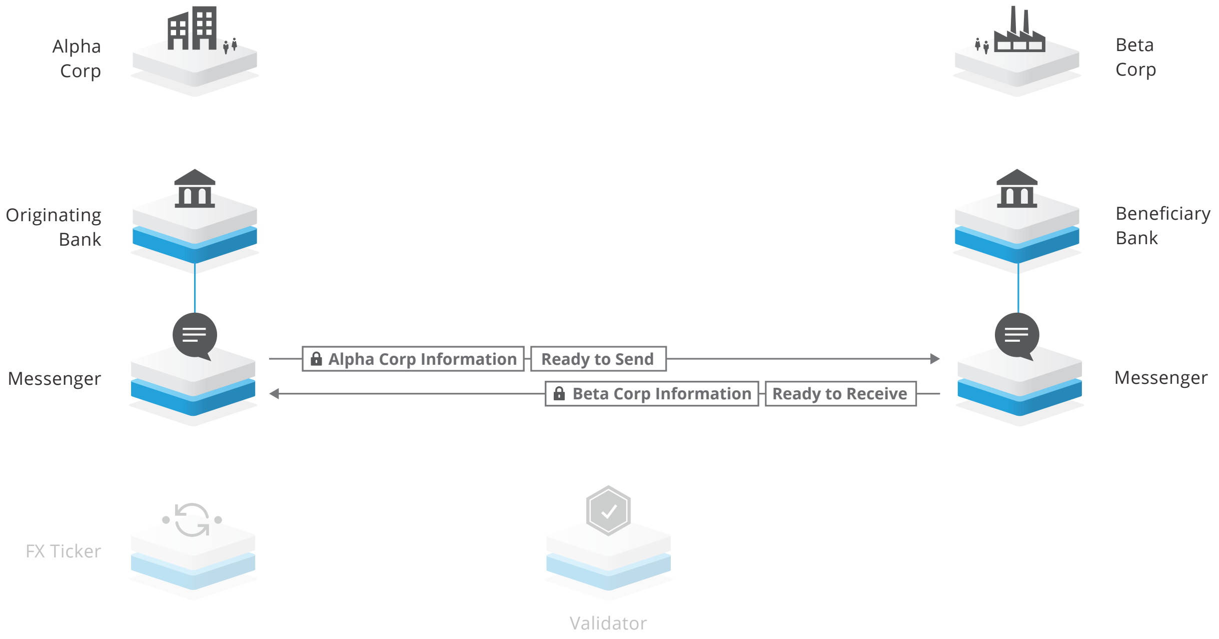
ID, which is included in the `Accept Quote` response. The beneficiary bank reviews the quote and performs compliance checks to ensure that:

a. The payment terms are acceptable.

b. The additional payment information requested from the originating bank is present and sufficient to process the payment.

6. If the terms and additional payment information are acceptable, the beneficiary bank makes a `Lock Quote` request. A locked quote indicates that both parties intend to process the payment and deliver the funds as described in the contract fields of the payment. The contract fields cannot be changed after the quote is locked.

7. Messenger instance at the originating bank receives a notification that the payment is now locked and updates the payment state in its own database to reflect the new state.



Both institutions now have an identical payment in a locked state with all the information that both institutions need to execute the payment.

PAYMENT PROCESS

After both banks accept the quote, the originating bank can initiate the end-to-end payment, which is comprised of three sub-payments:

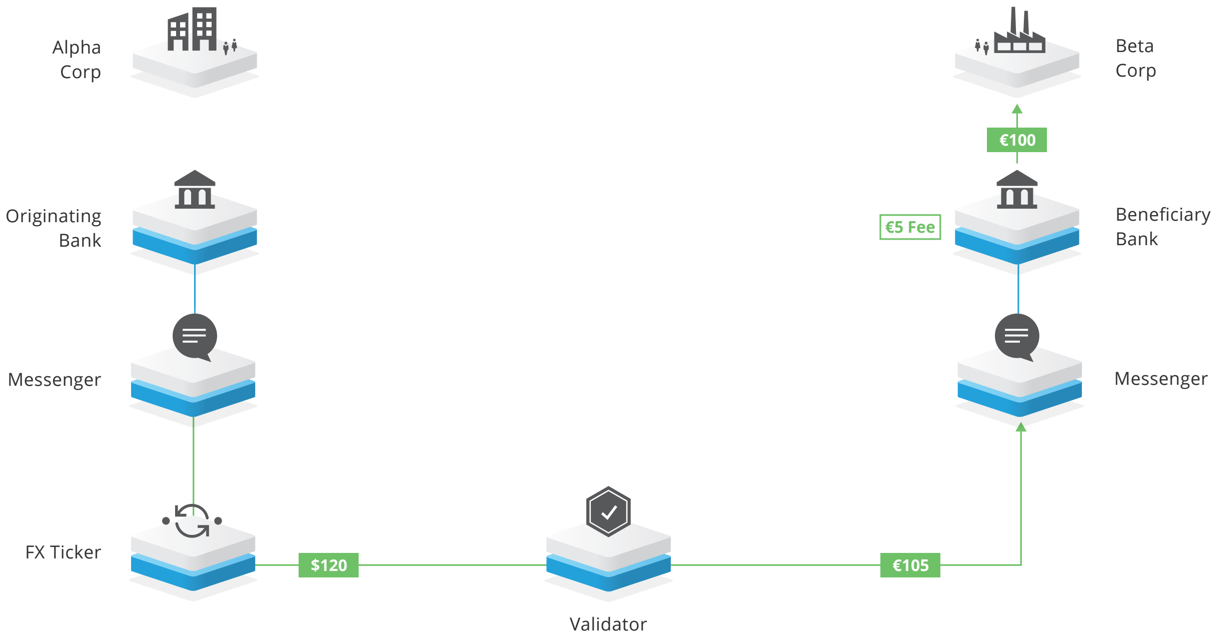
- **Sending payment:** The internal book transfer at the originating bank. The originating bank debits the originator's account and credits its own segregated account. Any fees charged by the originating bank are deducted from the originator's account in this transfer.
- **Settlement payment:** The transfers executed over the Interledger Protocol. The funds are transferred from the originating bank's transactional account (on its ILP Ledger) to the beneficiary bank's transactional account (on its ILP Ledger). This transfer is triggered automatically when the originating bank makes the `Submit Sub-Payment` request. Executing the settlement payment does not require any additional action on the part of the originating bank.
- **Receiving payment:** The internal book transfer at the beneficiary bank. The beneficiary bank debits its own segregated account and credits the beneficiary's account. Any fees charged by the beneficiary bank are deducted from the beneficiary's account in this transfer.

Executing the end-to-end payment involves the following steps:

1. The originating bank makes an internal book transfer debiting the funds from the sender's account. In the example above, $125 is deducted: $120 for the payment and $5 for the originating bank's fee.
2. The originating bank makes a `Submit Sending Payment` request to Messenger to acknowledge that the funds have been debited from the originator's account in the bank's internal systems. The request to Messenger does not affect the bank's internal systems. Typically, integrationlogic coordinates the transfer in the internal systems with executing the `Submit Sending Payment` request.
3. The originating bank's Messenger instance notifies the beneficiary bank's Messenger instance that the funds have been debited from the sender's account.
4. The `Submit Sending Payment` request triggers the settlement payment, which transfers the funds through the Interledger Protocol (ILP) from the originating bank's ILP Ledger to the beneficiary bank's ILP Ledger.
5. The originating bank's Messenger instance notifies the beneficiary bank's Messenger instance that the settlement payment has been sent.
6. The beneficiary bank sees that the ILP transfers have been validated by Validator and makes an internal book transfer to deliver the funds to the beneficiary's account.
7. The beneficiary bank makes a `Submit Receiving Payment` request to its Messenger instance, which changes the state of the payment to `succeeded` in its database.

8. The beneficiary bank's instance notifies the originating bank's Messenger instance that the funds have been delivered to the beneficiary’s account.
9. After receiving the notification, the originating bank's Messenger instance changes the state of the payment to `succeeded` in its database.

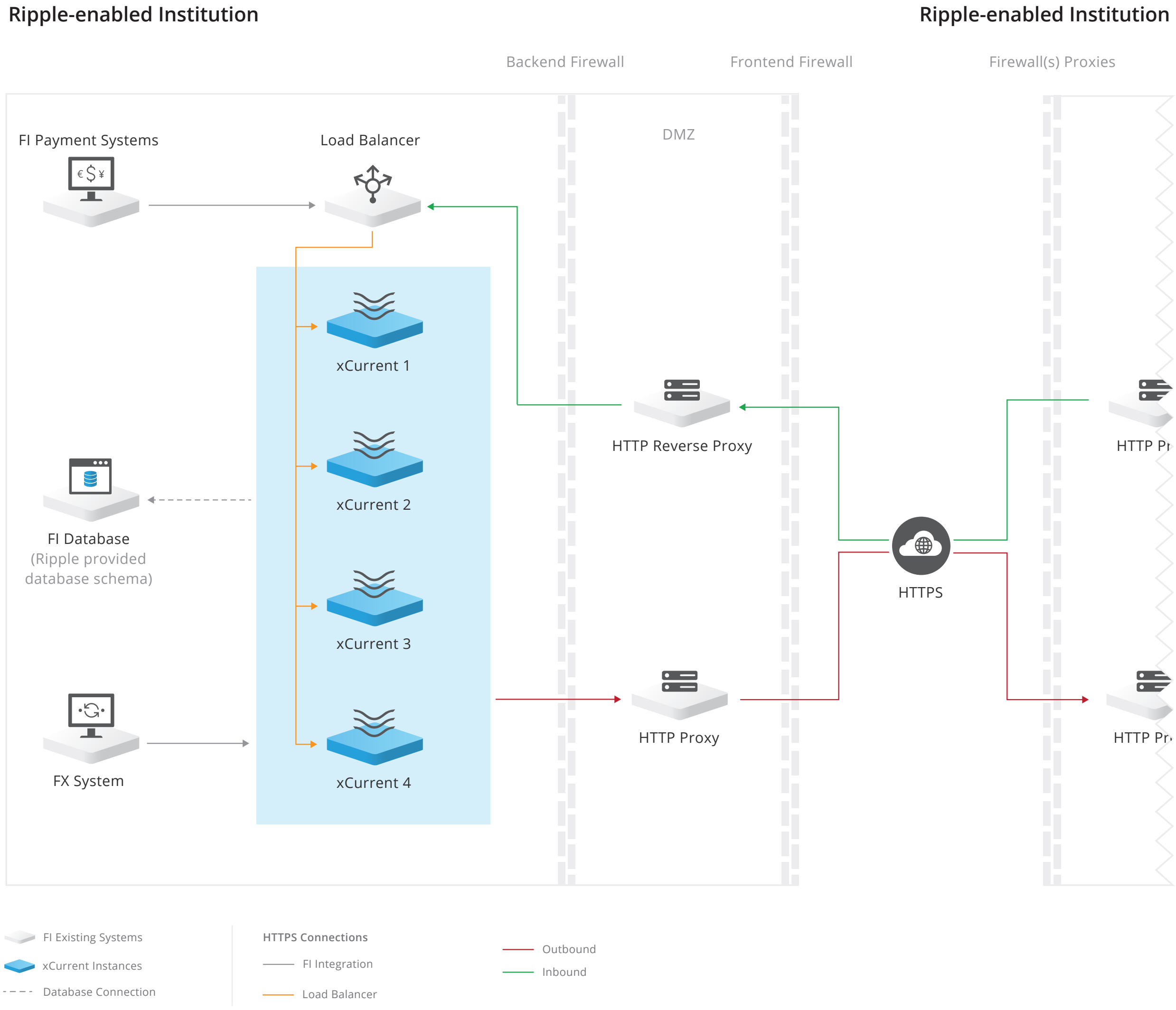
At this point, both parties consider the payment complete.



The payment is now complete. Alpha Corp sent $125 USD and €100 EUR was delivered to Beta Corp’s account.

Reference Architecture

xCurrent is typically installed on-premises behind the corporate firewall of a bank, with a load balancer handling inbound connections to Messenger and a proxy server handling inbound and outbound connections to all xCurrent components. Banks can deploy multiple instances of the xCurrent behind the load balancer to scale to the volume of payments. A typical production deployment of xCurrent at a bank is depicted in the following example:



Note: If a bank is not providing liquidity, only Messenger and ILP Ledger must be configured.

The above diagram includes the following features:

- Both Ripple-enabled institutions can send and receive payments through xCurrent.
- The originating or beneficiary institution can provide liquidity.
- xCurrent is deployed in a secure, trusted network zone, behind corporate firewalls and the DMZ. (xCurrent should not be deployed inside the DMZ.)
- All components of xCurrent are co-located on a single application server and communicate with each other over HTTPS and use TLS certificates for authentication.
- xCurrent deployment communicates with partner xCurrent deployments over HTTPS and use TLS certificates for authentication.
- xCurrent supports load balancing and horizontal scaling for all services to create an active-active, highly available (HA) deployment.
- The databases for Messenger, ILP Ledger, Validator and FX Ticker are created with the provided database schemas and deployed on the same database server.
- Each instance of each xCurrent component (Messenger, ILP Ledger, Validator and FX Ticker) requires access to a database. It is supported to use separate database instances for each component or a single database instance for all the components of xCurrent.
  Ripple recommends the latter option.

Technical Requirements

Operating System	Red Hat Enterprise Linux (RHEL) 6.7 and 7.2
Architecture	x86 (64-bit)
RAM	8 GB
CPU	4 Cores
Disk Storage	100 GB
Supported Database Connections	PostgreSQL 9.4 Microsoft® SQL Server® 2012 Microsoft® SQL Server® 2014
Deployment Options	RPM
RPM Dependencies	Node.js v6.9.0 or later

About Ripple

Ripple provides one frictionless experience to send money globally using the power of blockchain. By joining Ripple's growing, global network, financial institutions can process their customers' payments anywhere in the world instantly, reliably and cost-effectively. Banks and payment providers can use the digital asset XRP to further reduce their costs and access new markets.

With offices in San Francisco, New York, London, Sydney, Mumbai and Luxembourg, Ripple has more than 90 customers around the world.

Contact Us

To learn about joining RippleNet and leveraging xCurrent for cross-border payments, please contact us at ripple.com/contact.

ripple