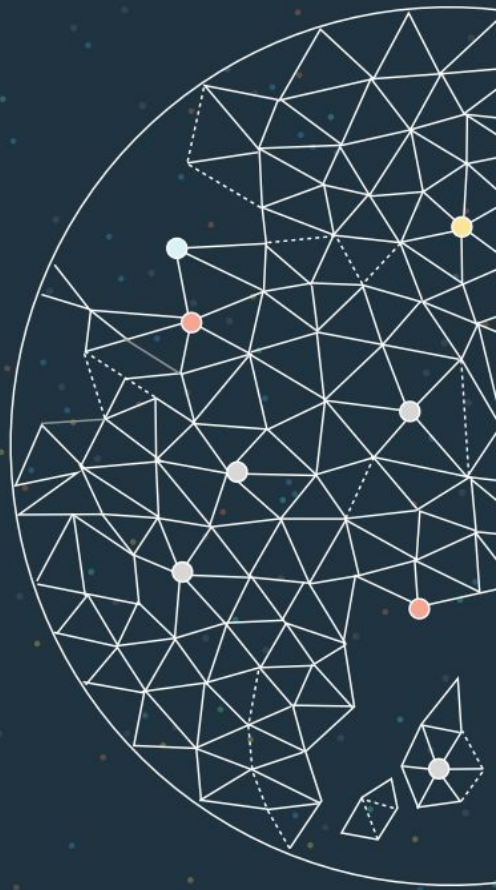
stellar

Creating the Internet for Payments

@jedmccaleb, Co-Founder of @stellarorg



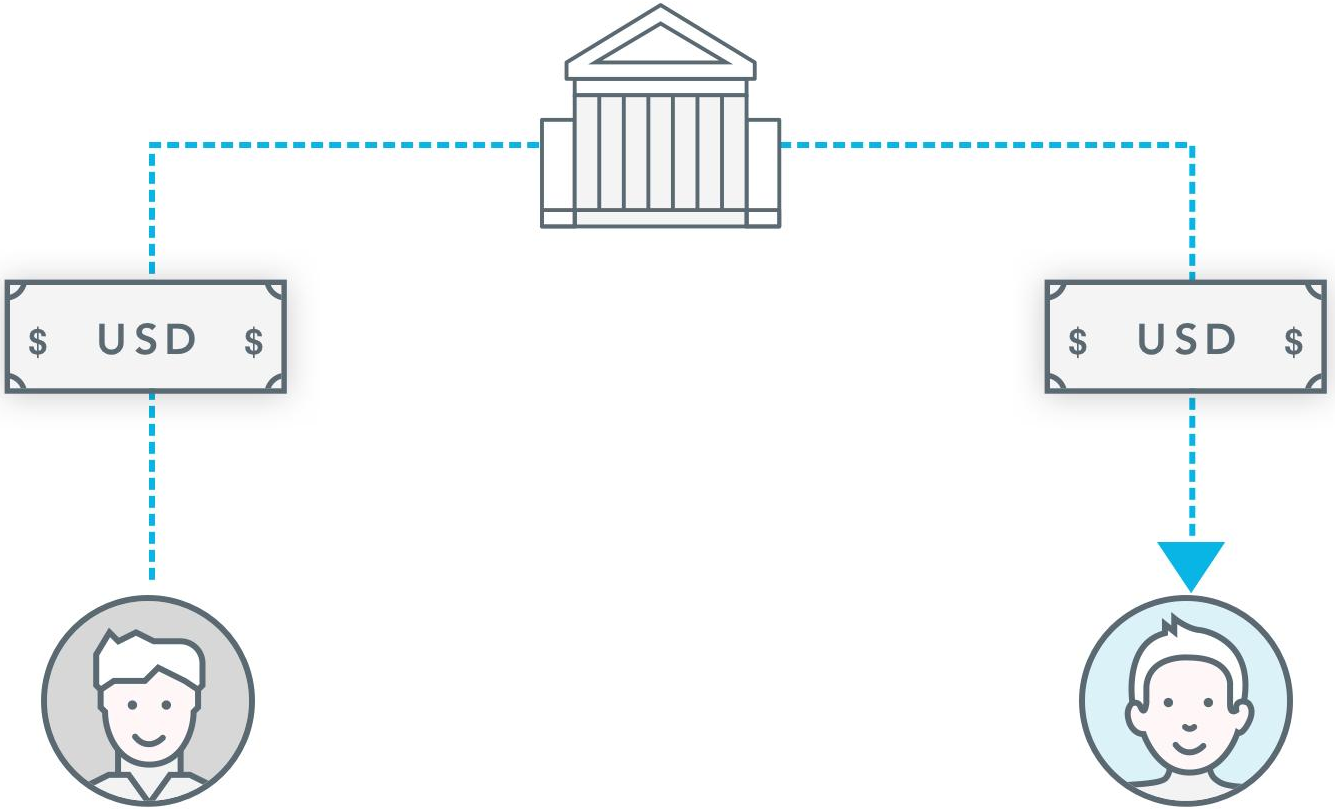
Payments are broken

Sending 50 yuan to Thailand

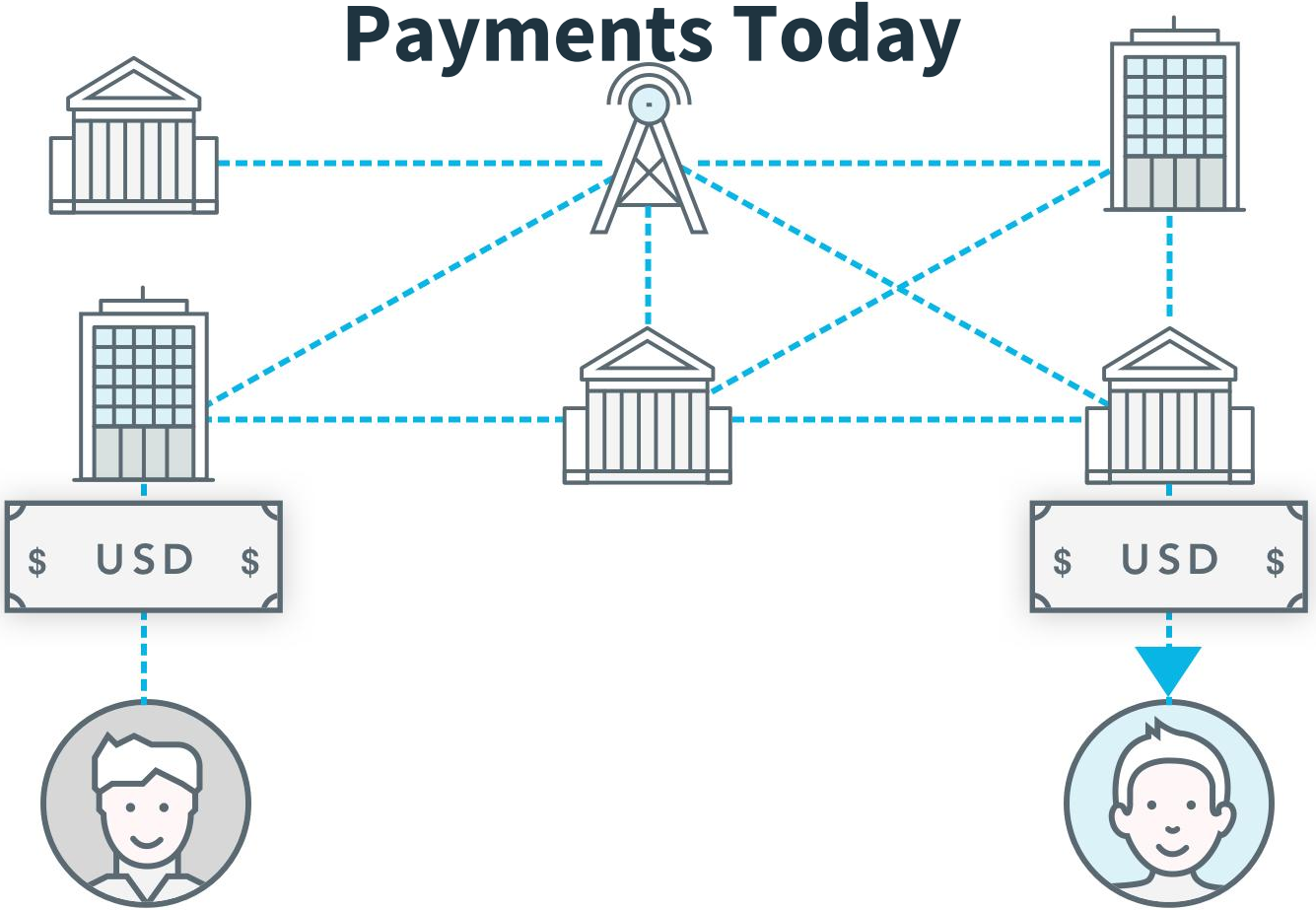
Receiving payments from Nigeria

Payments Today

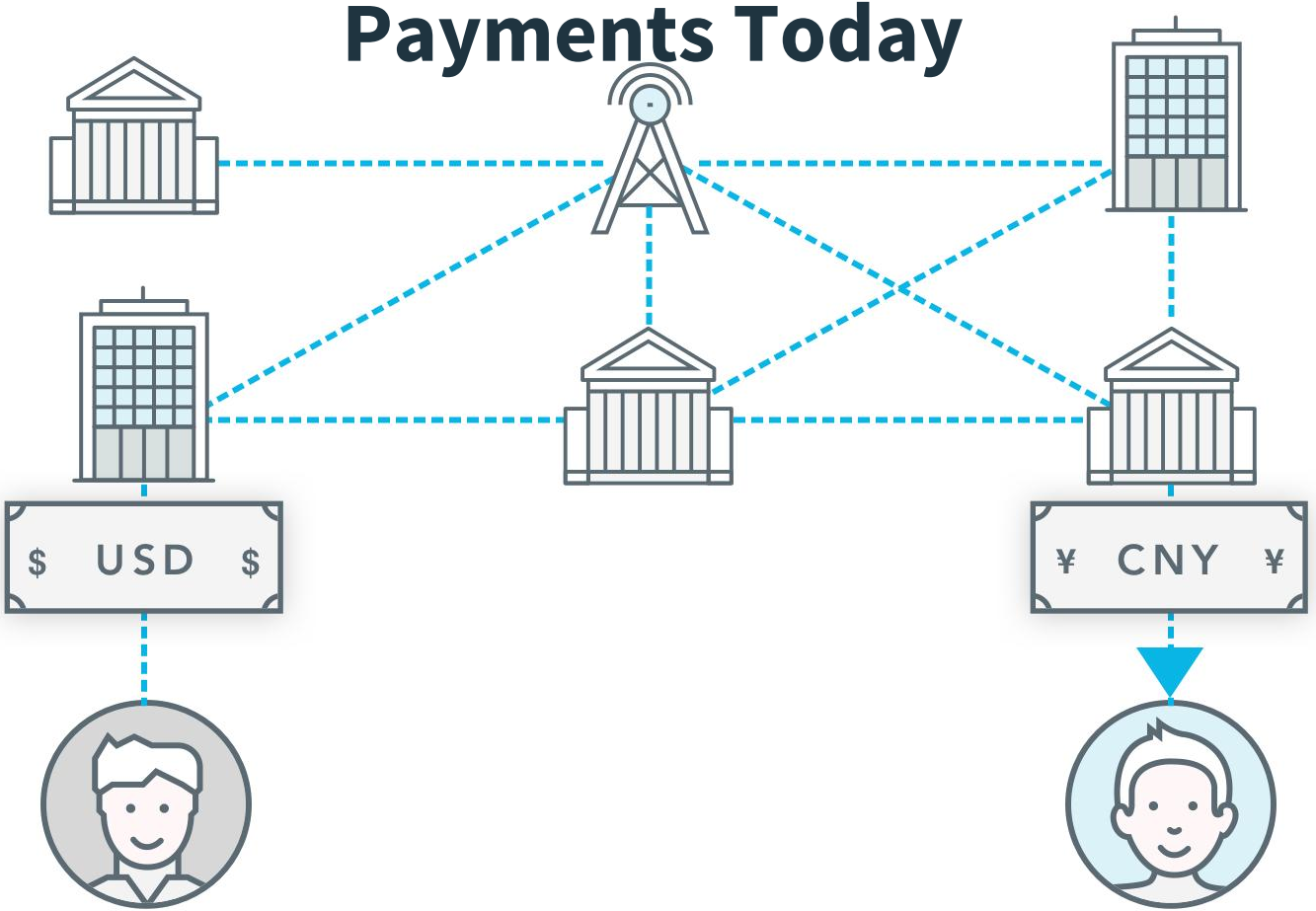
Payments Today



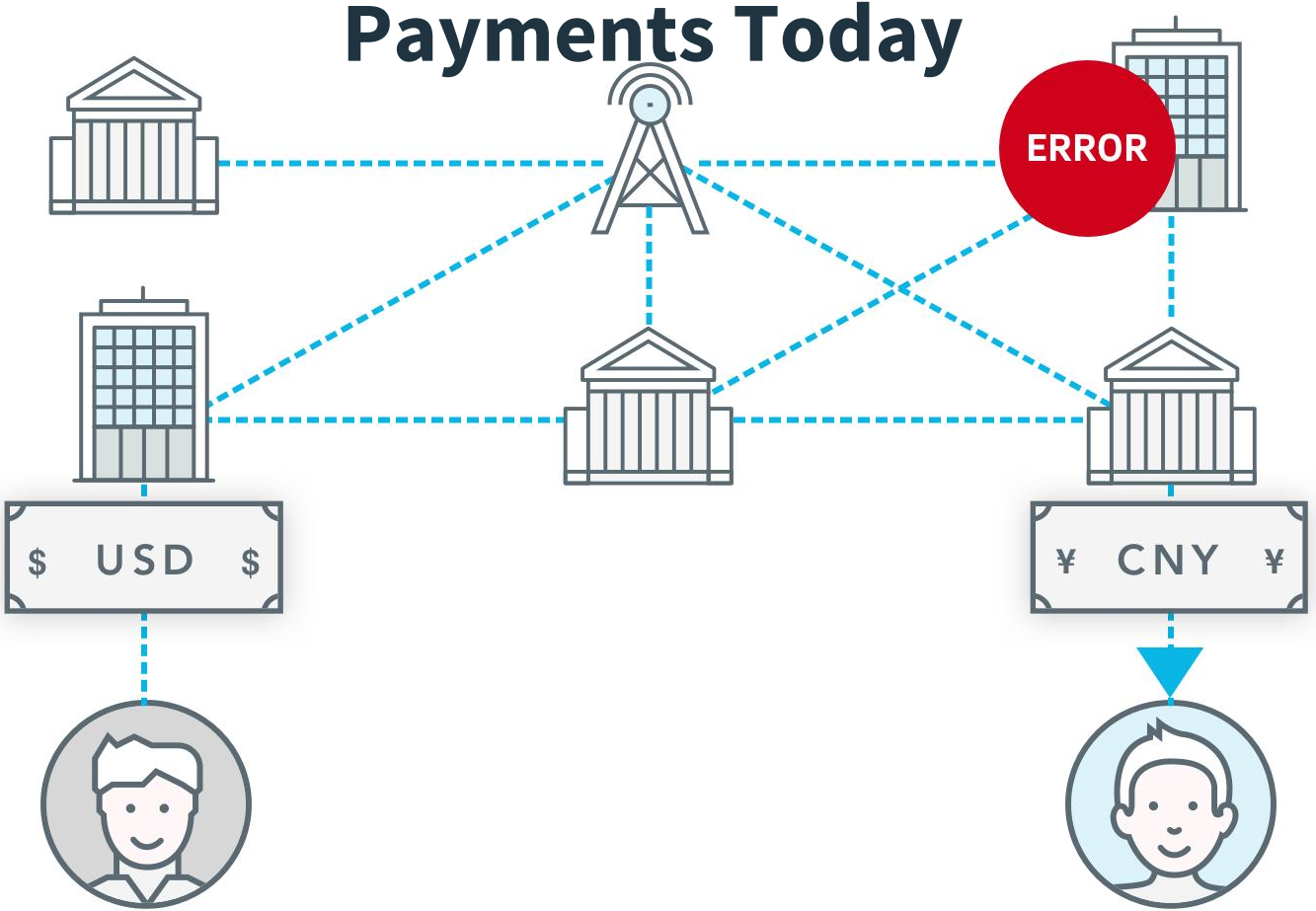
Payments Today



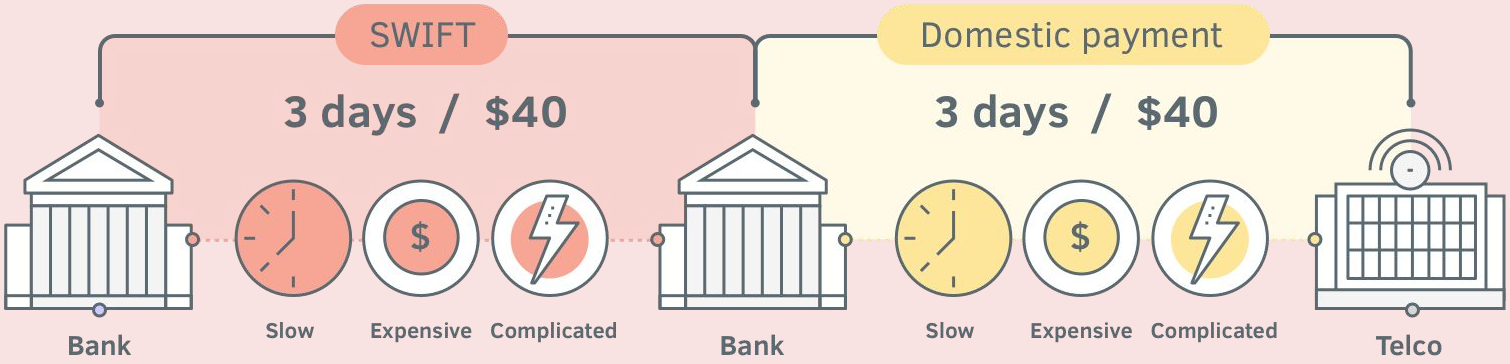
Payments Today



Payments Today



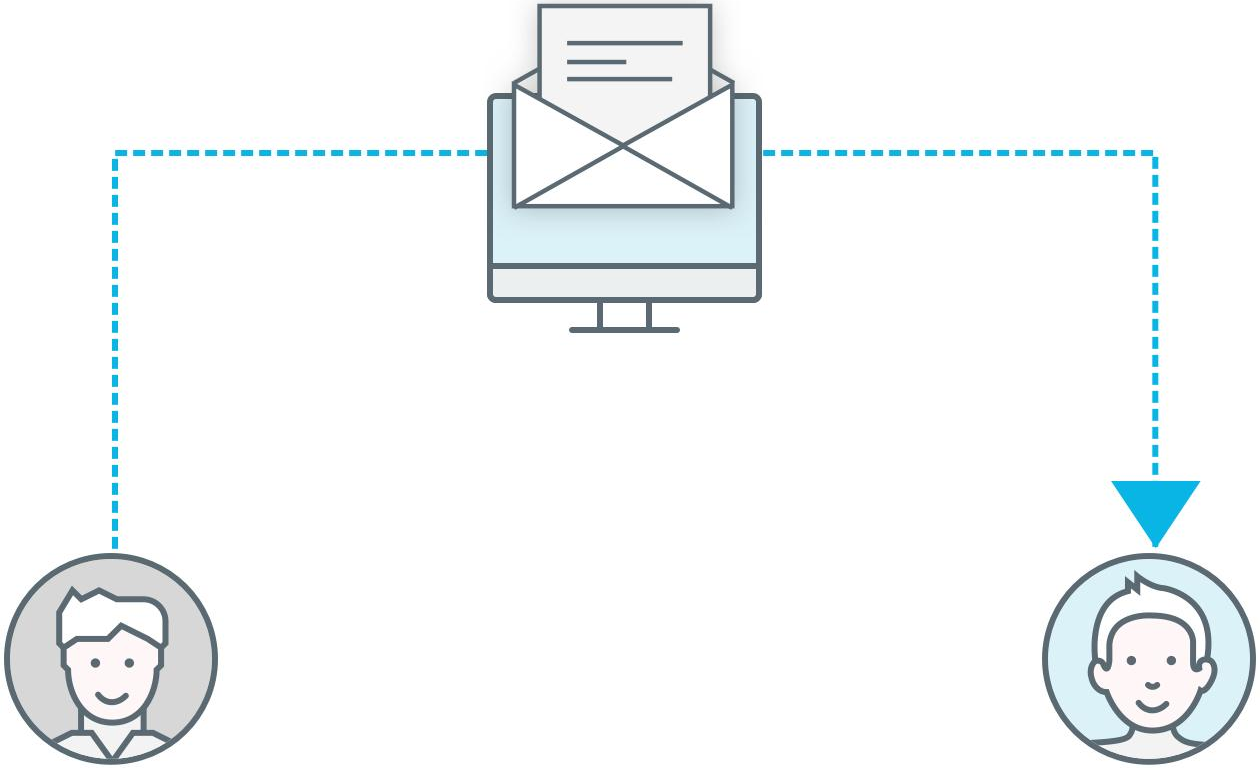
Payment Networks today



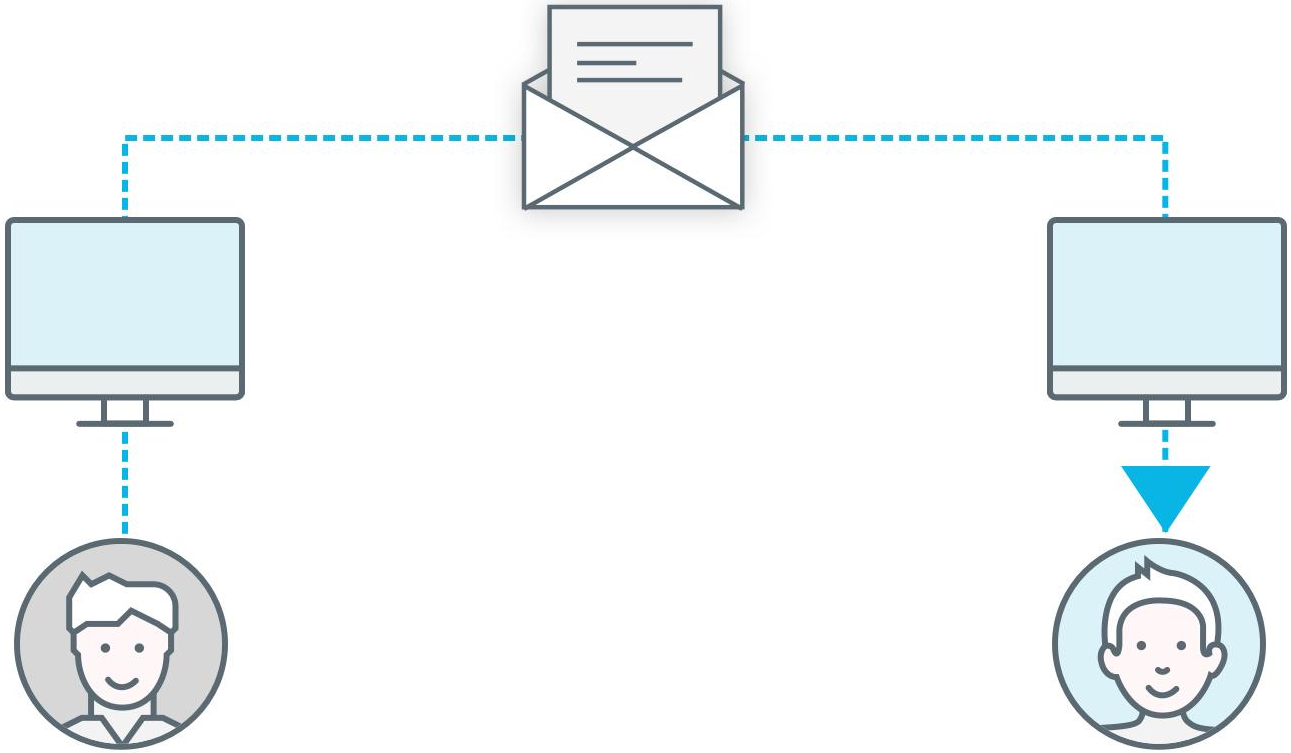
- Payments are slow and expensive
- Compliance is burdensome and risky
- Lack of interoperability

Is there a way to fix this?

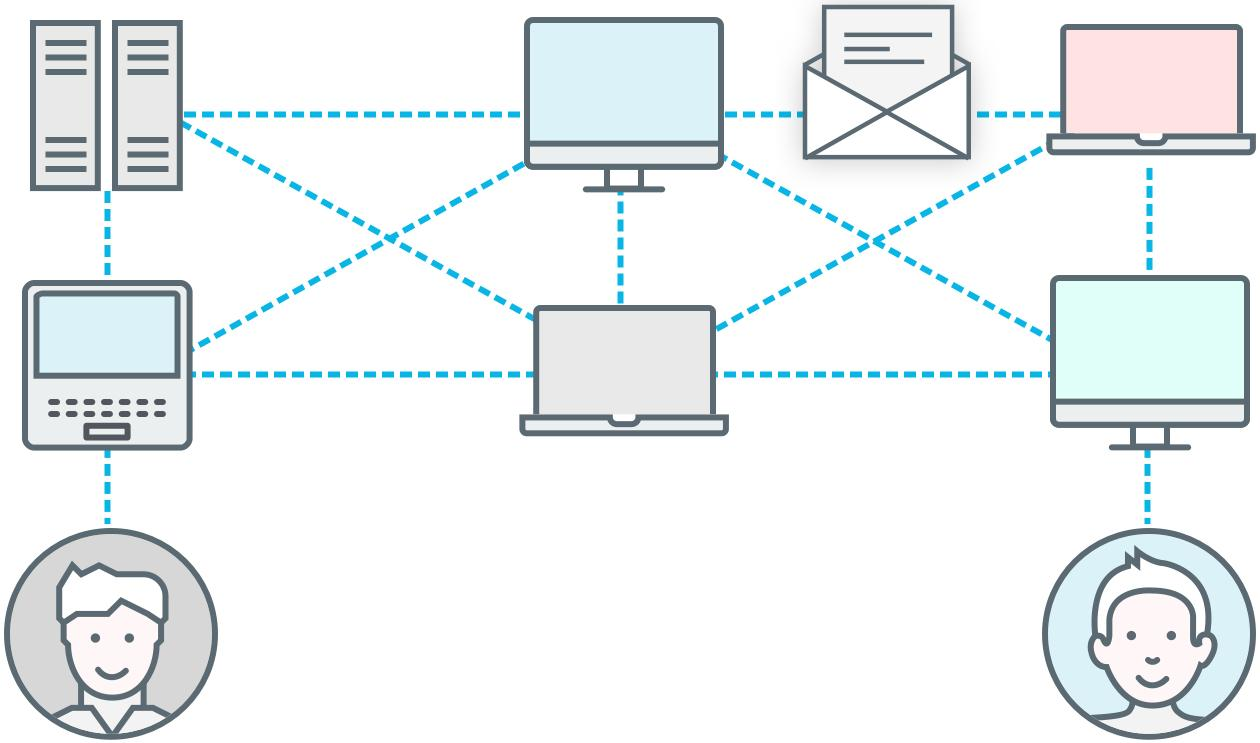
Analogy: email



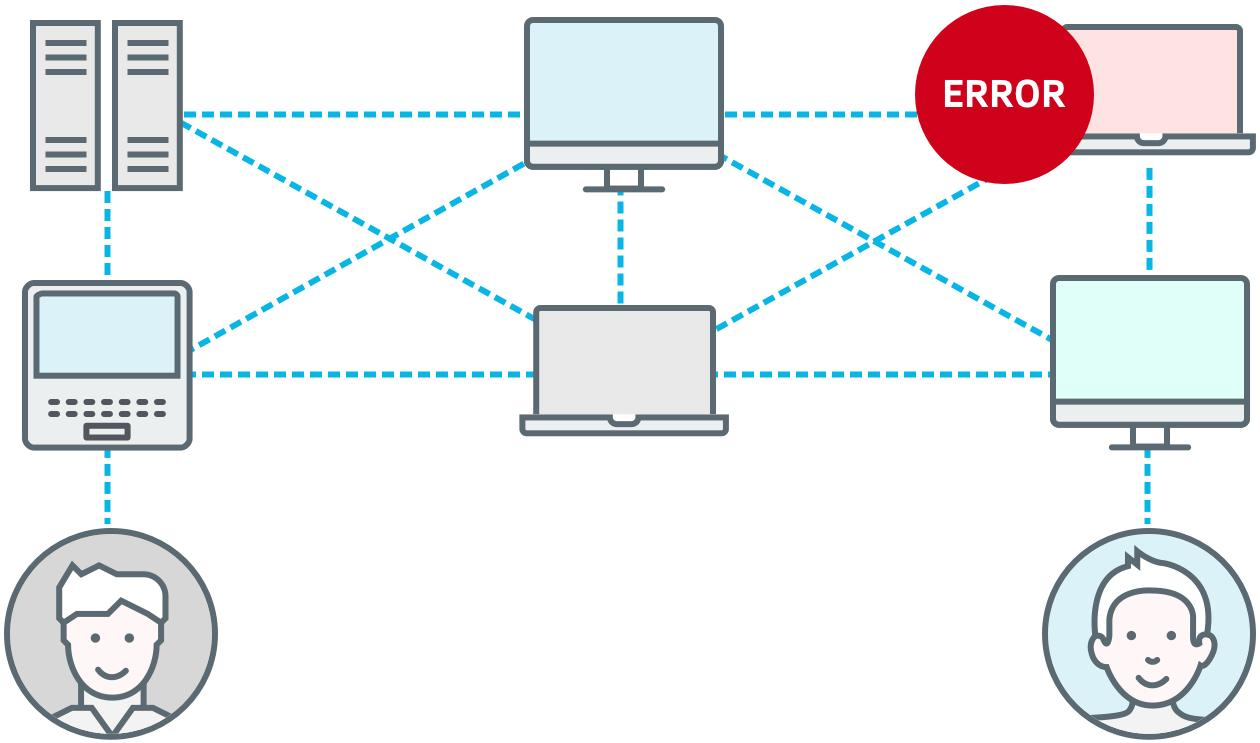
Analogy: email



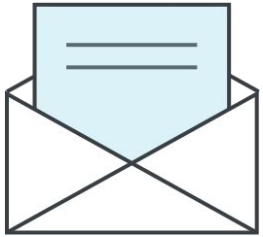
Analogy: email



Analogy: email



SMTP



- Open
- Interoperable, From/To anywhere
- Fast

Network Working Group
Request for Comments: DRAFT
Replaces: RFC 788, 780, 772

J. Postel
ISI
August 1982

SIMPLE MAIL TRANSFER PROTOCOL

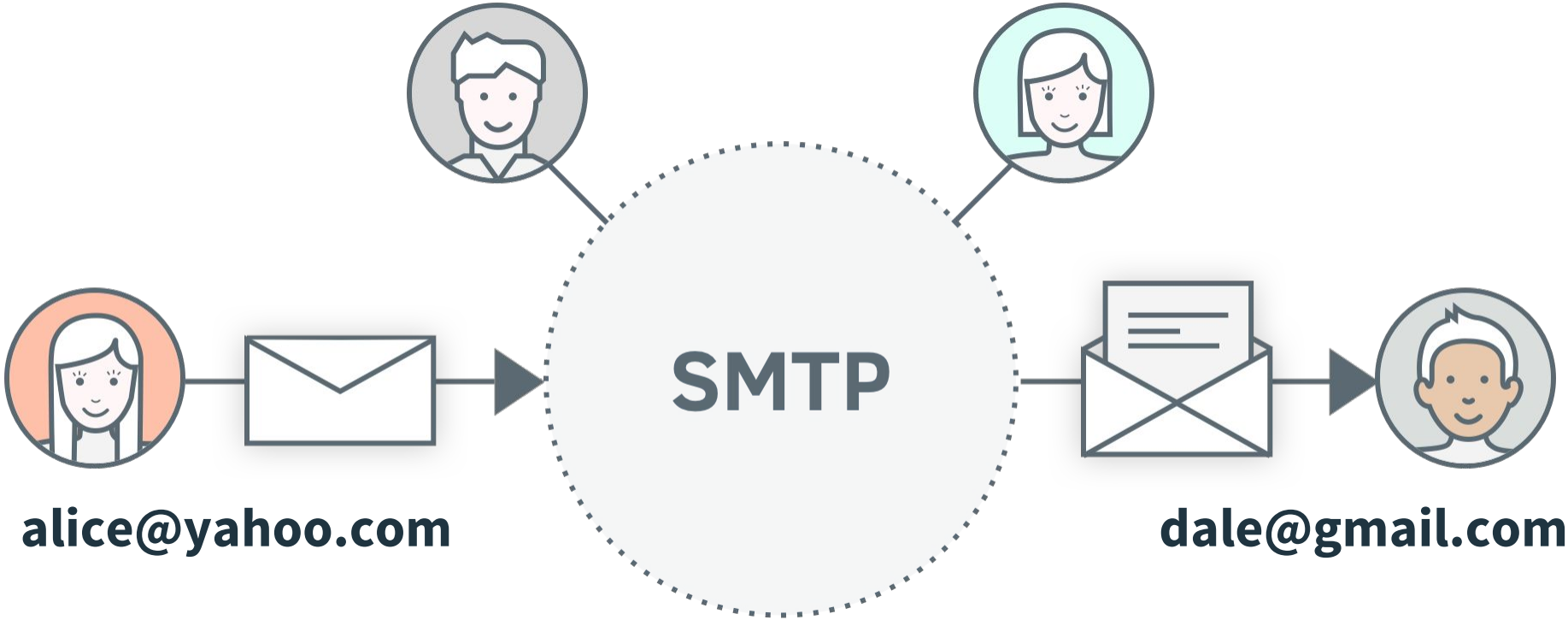
1. INTRODUCTION

The objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently.

SMTP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. Appendices A, B, C, and D describe the use of SMTP with various transport services. A Glossary provides the definitions of terms as used in this document.

An important feature of SMTP is its capability to relay mail across transport service environments. A transport service provides an interprocess communication environment (IPCE). An IPCE may cover one network, several networks, or a subset of a network. It is important to realize that transport systems (or IPCEs) are not one-to-one with networks. A process can communicate directly with another process through any mutually known IPCE. Mail is an application or use of interprocess communication. Mail can be communicated between processes in different IPCEs by relaying through a process connected to two (or more) IPCEs. More specifically, mail can be relayed between hosts on different transport systems by a host on both transport systems.

Open protocol for communications



Why can't we use the same model for payments?

1. **Store, prove possession and transmit value.**
2. Digitally represent existing currencies and assets.
3. Make payments interoperable across institutions and currencies.
4. And make sure your system still scales!

Why can't we use the same model for payments?

1. Store, prove possession and transmit value.
2. **Digitally represent existing currencies and assets.**
3. Make payments interoperable across institutions and currencies.
4. And make sure your system still scales!

Why can't we use the same model for payments?

1. Store, prove possession and transmit value.
2. Digitally represent existing currencies and assets.
3. **Make payments interoperable across institutions and currencies.**
4. And make sure your system still scales!

Why can't we use the same model for payments?

1. Store, prove possession and transmit value.
2. Digitally represent existing currencies and assets.
3. Make payments interoperable across institutions and currencies.
4. **And make sure your system still scales!**

Why can't we use the same model for payments?

1. Store, prove possession and transmit value.
2. Digitally represent existing currencies and assets.
3. Make payments interoperable across institutions and currencies.
4. And make sure your system still scales!

Is it possible to solve these problems?

Until one of these visionary
Satoshis gave us Bitcoin

Contributions of Bitcoin

- Converts a real world resource (electricity) into a digital asset
- Provides an immutable public record

But this still leaves these remaining problems

1. ~~Store, prove possession and transmit value.~~
2. **Digitally represent existing currencies and assets.**
3. **Make payments interoperable across institutions and currencies.**
4. **And make sure your system still scales!**

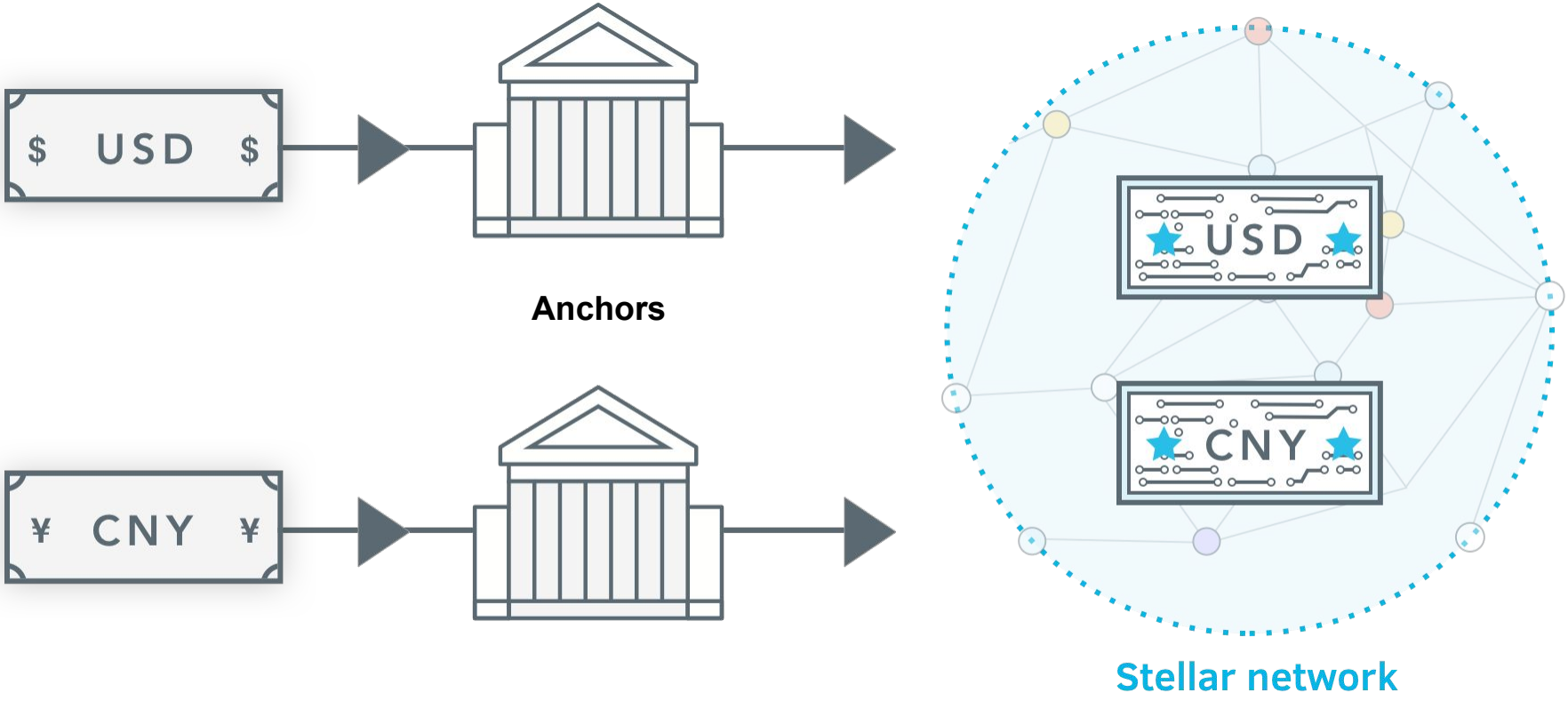
Stellar solves the 3 remaining problems

Problem 2:

Digitally represent existing currencies and assets.

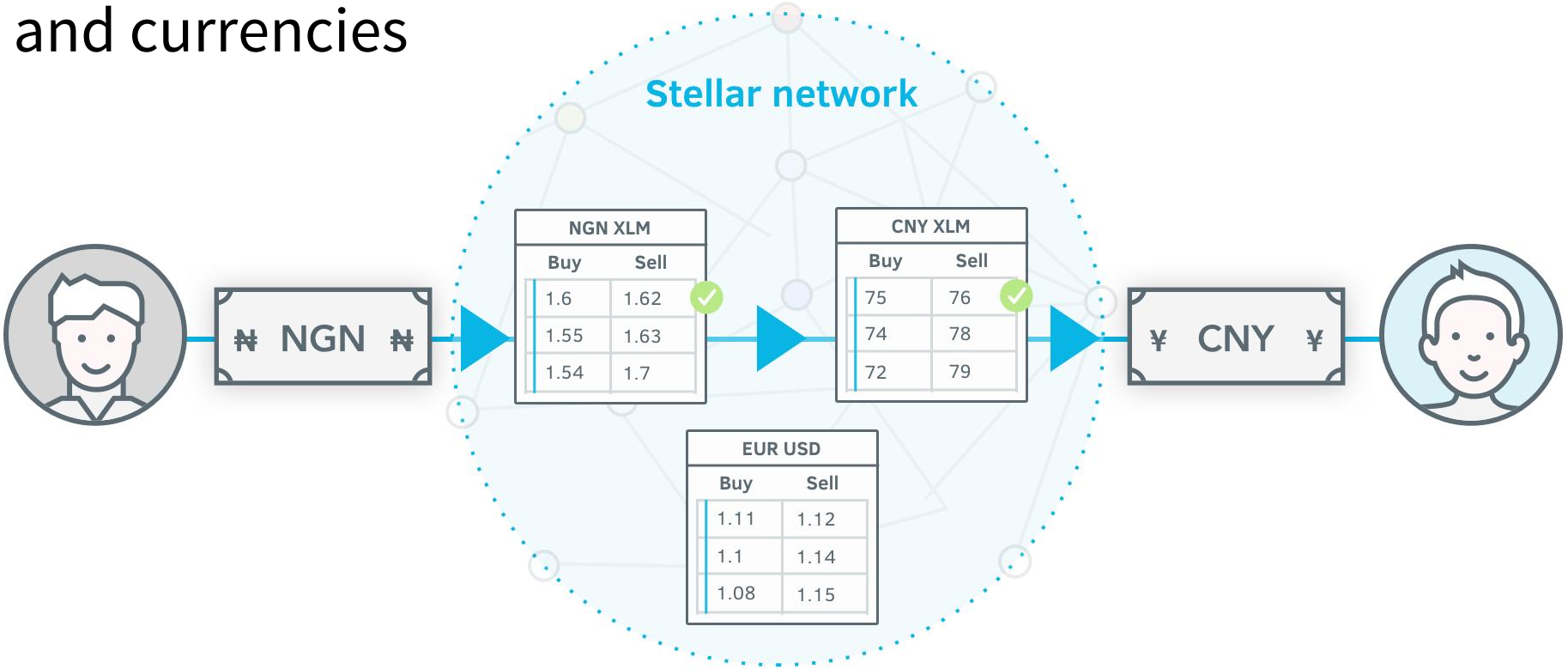
Problem 2:

Digitally represent existing currencies and assets.



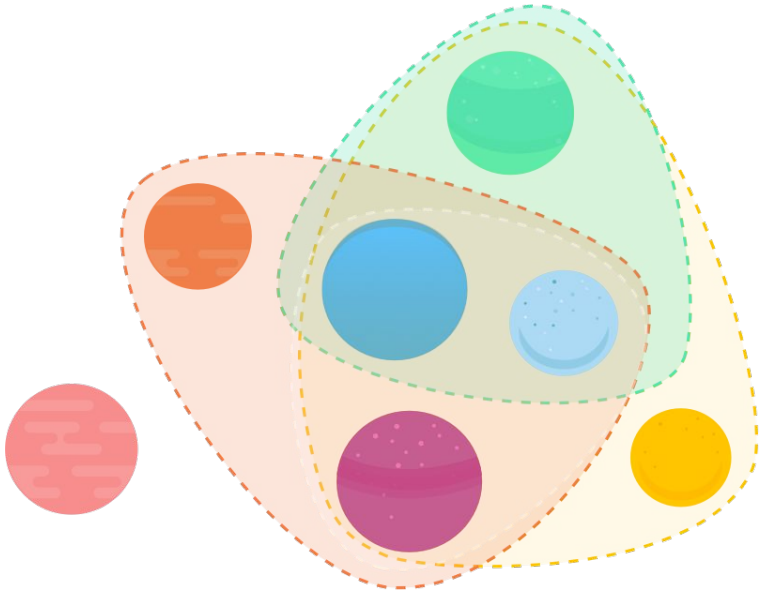
Problem 3:

Make payments interoperable across institutions and currencies



Problem 4:

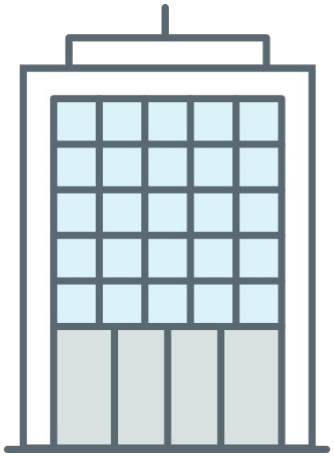
Make sure your system still scales



Stellar Consensus Protocol (SCP)

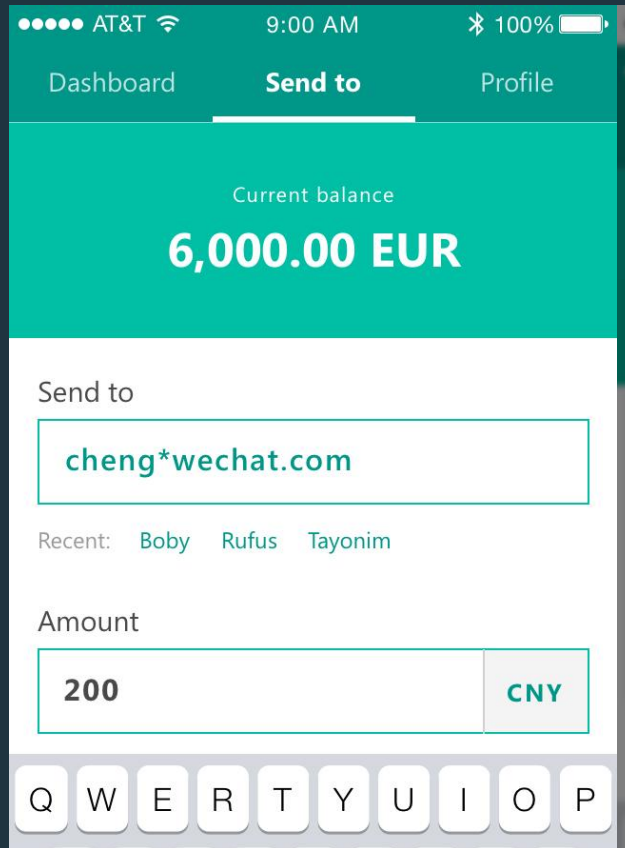
Payment Addresses

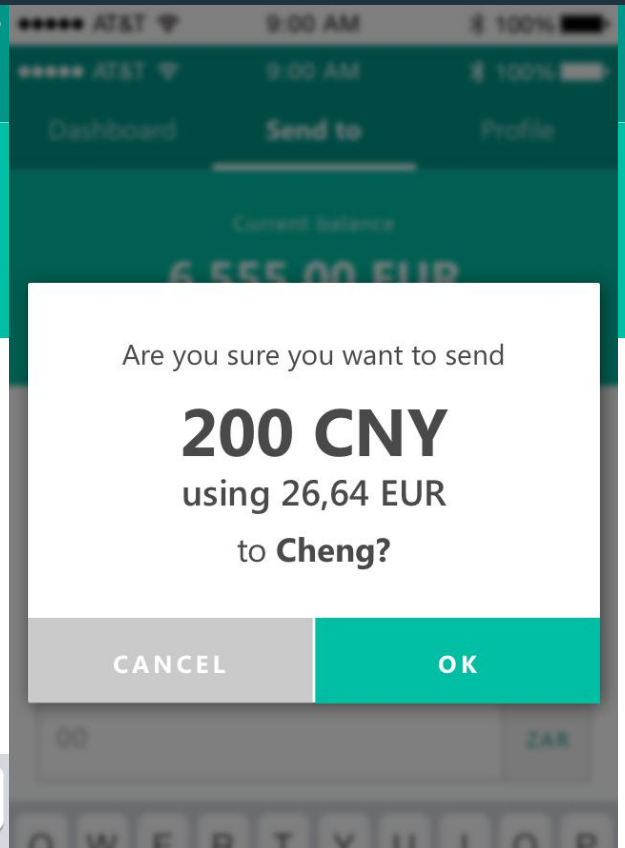
bob*barclays.com

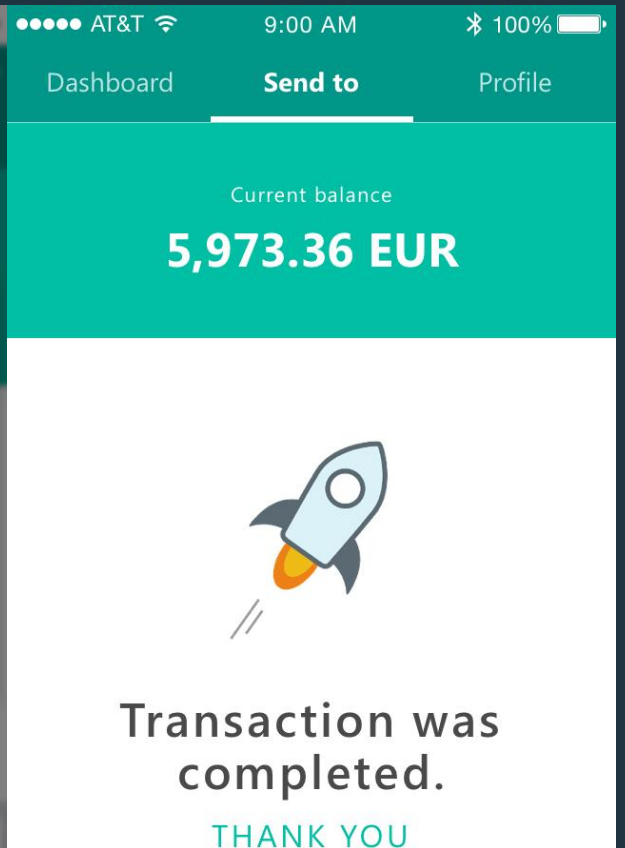


tunde*gtbank.com.ng

Bank's Mobile App

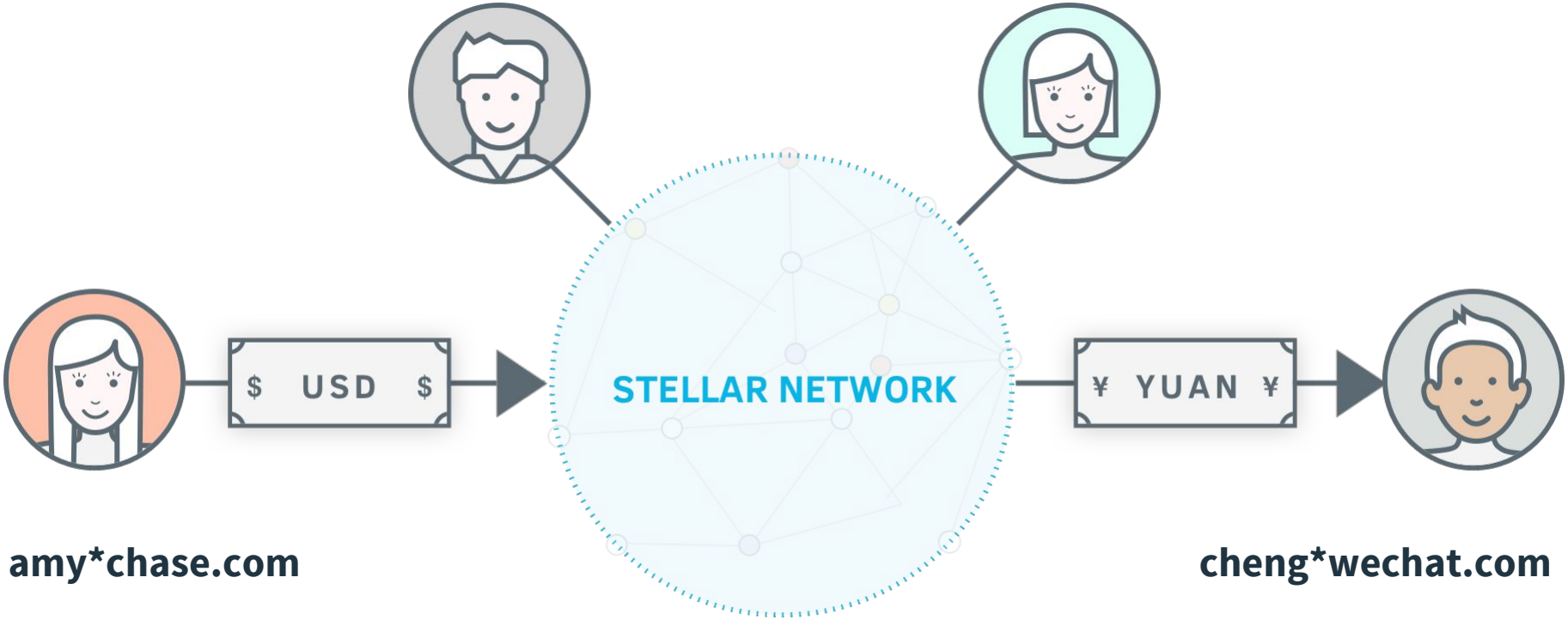






stellar

An open protocol for payments

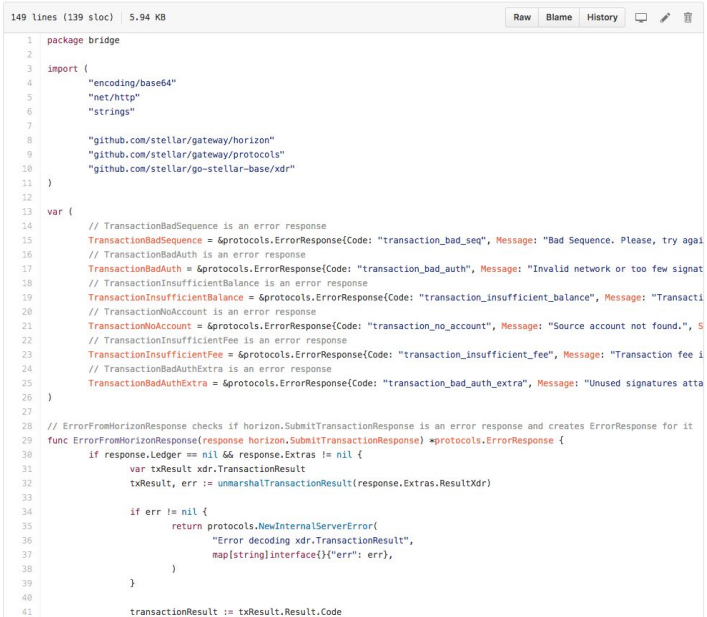


How does this become a standard?

Stellar Development Foundation

Stellar Development Foundation

Maintain the Code



149 lines (139 sloc) | 5.94 KB — Raw | Blame | History

```
package bridge

import (
        "encoding/base64"
        "net/http"
        "strings"

        "github.com/stellar/gateway/horizon"
        "github.com/stellar/gateway/protocols"
        "github.com/stellar/go-stellar-base/xdr"
)

var (
        // TransactionBadSequence is an error response
        TransactionBadSequence = &protocols.ErrorResponse{Code: "transaction_bad_seq", Message: "Bad Sequence. Please, try agai
        // TransactionBadAuth is an error response
        TransactionBadAuth = &protocols.ErrorResponse{Code: "transaction_bad_auth", Message: "Invalid network or too few signat
        // TransactionInsufficientBalance is an error response
        TransactionInsufficientBalance = &protocols.ErrorResponse{Code: "transaction_insufficient_balance", Message: "Transacti
        // TransactionNoAccount is an error response
        TransactionNoAccount = &protocols.ErrorResponse{Code: "transaction_no_account", Message: "Source account not found.", S
        // TransactionInsufficientFee is an error response
        TransactionInsufficientFee = &protocols.ErrorResponse{Code: "transaction_insufficient_fee", Message: "Transaction fee i
        // TransactionBadAuthExtra is an error response
        TransactionBadAuthExtra = &protocols.ErrorResponse{Code: "transaction_bad_auth_extra", Message: "Unused signatures atta
)

// ErrorFromHorizonResponse checks if horizon.SubmitTransactionResponse is an error response and creates ErrorResponse for it
func ErrorFromHorizonResponse(response horizon.SubmitTransactionResponse) *protocols.ErrorResponse {
        if response.Ledger == nil && response.Extras != nil {
                var txResult xdr.TransactionResult
                txResult, err := unmarshalTransactionResult(response.Extras.ResultXdr)

                if err != nil {
                        return protocols.NewInternalServerError(
                                "Error decoding xdr.TransactionResult",
                                map[string]interface{}{"err": err},
                        )
                }

                transactionResult := txResult.Result.Code
```

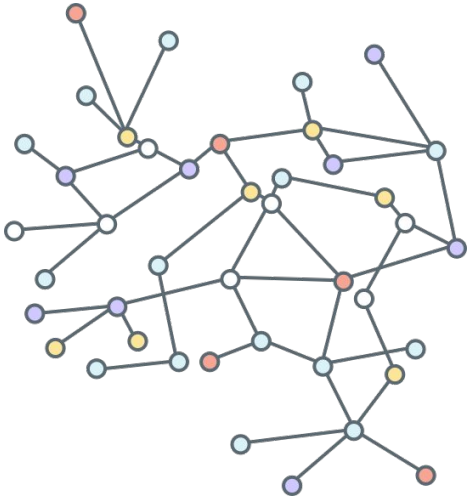
Stellar Development Foundation

Maintain the Code

149 lines (139 sloc) | 5.94 KB — Raw | Blame | History

```go
package bridge

import (
	"encoding/base64"
	"net/http"
	"strings"

	"github.com/stellar/gateway/horizon"
	"github.com/stellar/gateway/protocols"
	"github.com/stellar/go-stellar-base/xdr"
)

var (
	// TransactionBadSequence is an error response
	TransactionBadSequence = &protocols.ErrorResponse{Code: "transaction_bad_seq", Message: "Bad Sequence. Please, try agai
	// TransactionBadAuth is an error response
	TransactionBadAuth = &protocols.ErrorResponse{Code: "transaction_bad_auth", Message: "Invalid network or too few signat
	// TransactionInsufficientBalance is an error response
	TransactionInsufficientBalance = &protocols.ErrorResponse{Code: "transaction_insufficient_balance", Message: "Transacti
	// TransactionNoAccount is an error response
	TransactionNoAccount = &protocols.ErrorResponse{Code: "transaction_no_account", Message: "Source account not found.", S
	// TransactionInsufficientFee is an error response
	TransactionInsufficientFee = &protocols.ErrorResponse{Code: "transaction_insufficient_fee", Message: "Transaction fee i
	// TransactionBadAuthExtra is an error response
	TransactionBadAuthExtra = &protocols.ErrorResponse{Code: "transaction_bad_auth_extra", Message: "Unused signatures atta
)

// ErrorFromHorizonResponse checks if horizon.SubmitTransactionResponse is an error response and creates ErrorResponse for it
func ErrorFromHorizonResponse(response horizon.SubmitTransactionResponse) *protocols.ErrorResponse {
	if response.Ledger == nil && response.Extras != nil {
		var txResult xdr.TransactionResult
		txResult, err := unmarshalTransactionResult(response.Extras.ResultXdr)

		if err != nil {
			return protocols.NewInternalServerError(
				"Error decoding xdr.TransactionResult",
				map[string]interface{}{"err": err},
			)
		}

		transactionResult := txResult.Result.Code
```

Grow the Network



Deloitte.

stripe

ICICI Bank

coins.ph

PARKWAY
PROJECTS

LOBSTR

rehive

TEMPO
MONEY TRANSFER

Splash

Mifos

oradian

GTBank

Flutterwave®

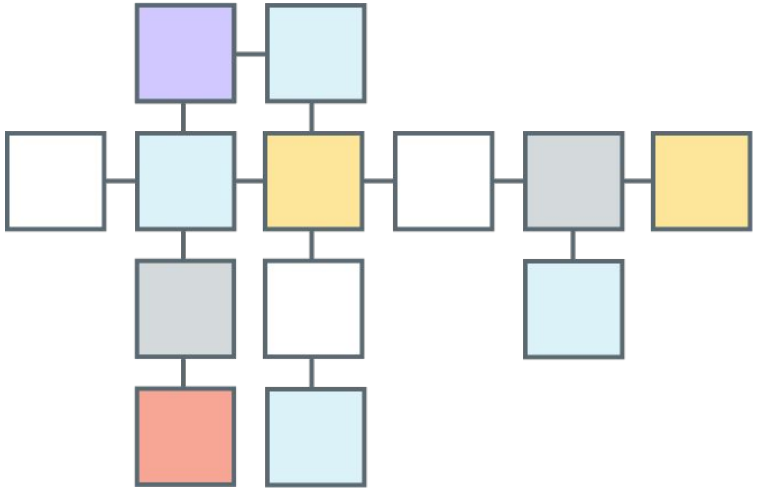
codetip.io

SIKA

Keystone
Bank

PRAEKELT

You are at the forefront of change

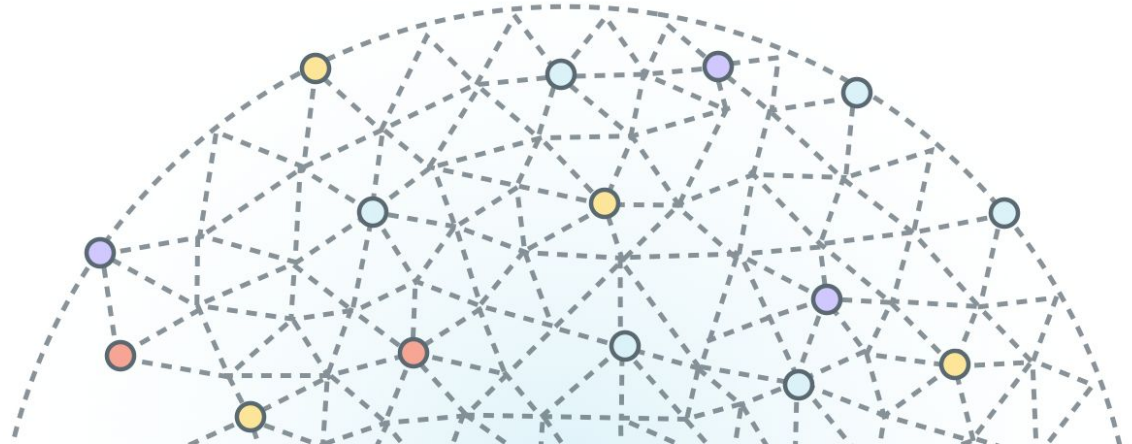


Implications of a Stellar enabled world

What people build with open platforms

What people build with open platforms

stellar

If we made you curious, ask at:

partners@stellar.org