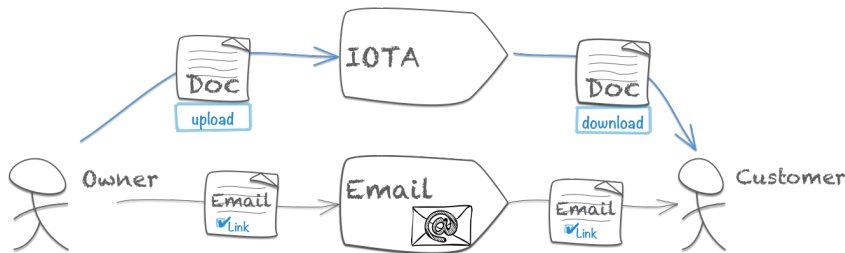# IOTA/FileExchange

Problem statement

What is the problem I'am trying to solve?

I want to exchange secure documents over email+iota

- to reduce letter shipping costs for companies
- to reduce paper wast for the end customer (and the earth)
- to reduce portal-login-hell for the end customer

> 'portal-login-hell' is the fact that a lot of companies have a portal with login where its documents are shared to the customer. This portals needs a own login. The end customer needs to handle this portal links+logins - the portal-login-hell.

Market description

What does the market look like? Where would you apply the use-case? Local? Regional? Worldwide?

> The market scope is worldwide every email user (theoretical)

B2C market

- big companies they have a lot of document exchange with end customer
- like banks and insurance companies
- they want to reduce its shipping costs

C2C market

- every email user who doesn't want share plain documents via email, slack or other untrusted services
- maybe 0,1% of the email user? - (its still a lot ~4 million user - [source])

## A detailed overview of my proposed solution

Classical mail is the way to exchange documents in traditional way. Where everybody trusts the shipping service.

Email could not disrupt this traditional mail business yet. Because its intrinsic unsecure. The email is not designed for. All efforts to close this gap failed so far (e.g. smime or pgp) - because the setup is to complex for the end customer.
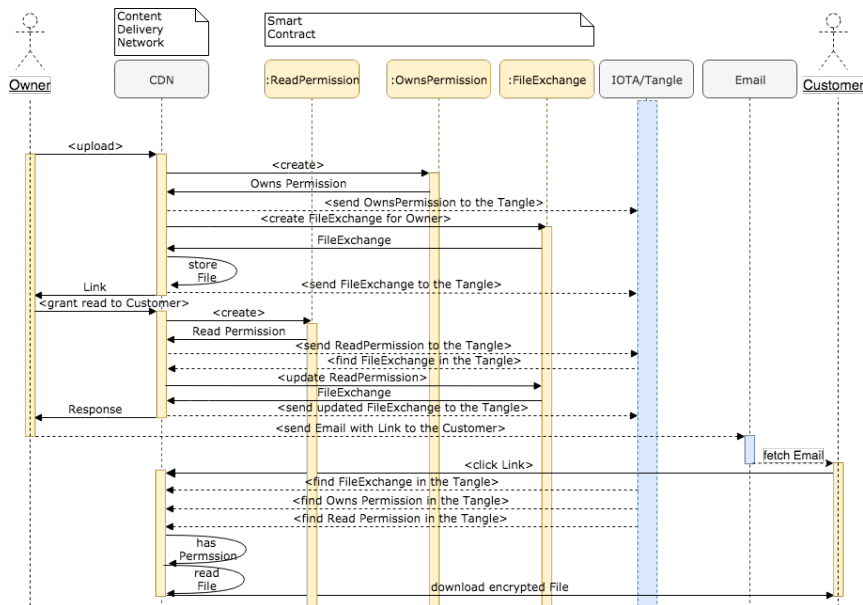
A lot of companies try to fits this gap through a own portal, where the customer get secure access to the documents from this company. These tends to portal-login-hell. I as customer must manage a lot of logins to different portals which I serve very rarely.

> I wants to fits this gap with email+iota.
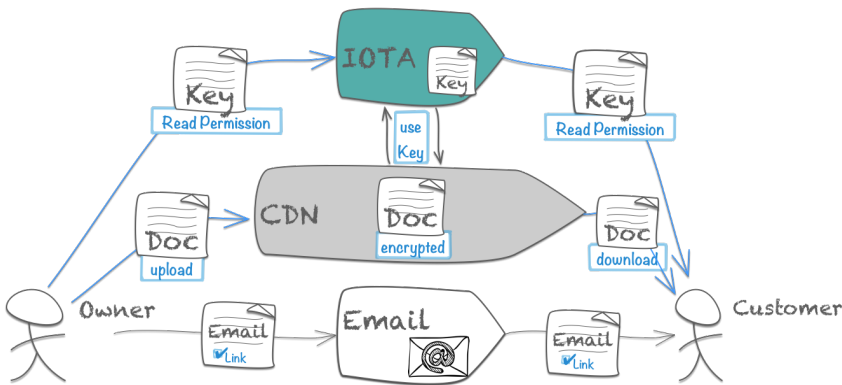
How IOTA is used in this use-case and what would be needed to achieve this solution.

IOTA is used to exchange permission and document information. Both (permissions and documents) are identifiable transactions in the tangle. They can be created, moved and managed with IOTA instruments.

Content Delivery Network

Smart Contract

Owner — CDN — :ReadPermission — :OwnsPermission — :FileExchange — IOTA/Tangle — Email — Customer

<upload>
<create>
Owns Permission
<send OwnsPermission to the Tangle>
<create FileExchange for Owner>
FileExchange
store File
Link
<send FileExchange to the Tangle>
<grant read to Customer>
<create>
Read Permission
<send ReadPermission to the Tangle>
<find FileExchange in the Tangle>
<update ReadPermission>
FileExchange
Response
<send updated FileExchange to the Tangle>
<send Email with Link to the Customer>
fetch Email
<click Link>
<find FileExchange in the Tangle>
<find Owns Permission in the Tangle>
<find Read Permission in the Tangle>
has Permssion
read File
download encrypted File

Imagin that IOTA serves a SmartContract:=FileExchange. For every uploaded File will a FileExchange instance created in the tangle. The purpose of the FileExchange instance is permission handling and the documentation of the lifecycle (upload, permission grants, download, etc).

The FileExchange Contract

- has an identity (iota-token)
- has an resource (link to encrypted byte stream)
- has an symetric key (aes) to en/decrypt the content
- has 2+ permissions - (1) *own-permission* from the user who upload the $FILE_{TO}UPLOAD$. (2) *read-permission* from the user who wants to download the document

The permissions are RAS key pairs (private and public). A permission

- have also an identity (iota-token).
- they can be moved between users over the tangle.

The detailed process is:

```
1 – A Owner upload a document to Content Delivery Network (iota.file.exchange.CDN)
1.1 Create RSA key pair for the Owner to decrypt the aes key on iota.file.exchange
.CDN
1.2 Sent the RSA key pair to the owner ledger
1.3 Store the encrypted document
1.4 Response the link to the encrypted document to the owner
2 – The owner grant a read permission to a Customer
2.1 Create RSA key pair for the Customer to decrypt the aes key (on customer side)
2.2 Sent the RSA key pair to the customer ledger
3 – The Owner send a email to the customer with the document link
4 – Customer download the encrypted dokument
4.1 Customer fetch aes key (encrypted)
4.2 Decrypt encrypted aes key with customer private keys
5 – Customer decrypt encrypted document with decrypted aes key
```

Solutions and visions

Long term solution (Vison)

Every people use IOTA/FileExchange for secure document exchange direct or indirect ()like email/messenger today). There exists multiple clients for any device, platform, tool and browser to use the IOTA/FileExchange.

> IOTA/FileExchange is a brand.

Short term solution

Keep it simple for the end customer. No installation required. No ledger creation required. The end customer require only it's standard email client and a browser.

To keep it simple for the end customer should IOTA instruments behind a HTTP service. The user login into a global web service and use IOTA as a service indirect. The legers and transaction will be managed from the http service. Equal to the coin exchanges platforms today.

There is an 1:1 relation between iota ledgers and email addresses they use the service. One idea could be that:

- every email address represent a user
- every email address is a ledger on server side
- every user can use the service (upload and download documents)
- the user use oauth to get access to the HTTP service
- one first login the ledger must be created

Fazit

IOTA/FileExchange could change the world with a simple use case wich fits the wishes of the companies to save shipping costs and the whishes of customer to reduce paper waste.

Links

- [Slides](#)
- [Paper](#)
- [Code](#) (Kotlin)