

Yellow paper

Request Network

The concept of smart financial audits on the Blockchain

Table of Contents

| | |
|--|----|
| Introduction to the concept of smart audit | 1 |
| The Request model | 3 |
| Customers and income | 5 |
| Suppliers and charges | 9 |
| Intercompany transactions | 12 |
| Taxes and duties | 14 |
| Credit analysis and financing | 16 |
| The new issues related to the blockchain | 18 |
| Business development of Request: onboarding and transition | 18 |

Introduction to the concept of smart audit

Financial audit of companies has three objectives:

- Ensure adequate procedures are in place to achieve optimized results (internal management)
- Verify that the company is in compliance with the competent authorities
- Protect commercial players from the risk of fraudulent practices

From turnover to wages and taxes, a financial audit will go through all the transactions of a company. We categorize an audit today into several sections: equity, assets, banks, customers, income, suppliers, charges, salaries, taxes, intercompanies, financial charges and income.

SECTIONS - FINANCIAL AUDIT



All financial statements of a company are scrutinized: the **balance sheet**, **profit and loss** statement, and **cash-flow** statement. All financial transactions should be documented in the company's general ledger. Whether documented in manual form or with an electronic system, each transaction should be consistently recorded with enough information to identify who created it and for what purpose. Each should be stamped with the transaction date. Auditing a company financials is a long and painful process. From the data collection to the multiple reconciliations and the understanding of the accounts and recommendations, it is hard. Besides, the cost and inaccuracy of an audit are two big issues. It's expensive and the audit is never perfect as errors went unnoticed every year.

With a system like Request and the Blockchain possibilities, there is no doubt that **accounting and auditing practices will change** and never be the same. Thanks to the interoperability, immutability and decentralized nature of the Blockchain, we'll see the emergence of the triple-entry accounting system (the invoice becomes the transaction) and the time stamping system (basic blockchain function that permanently register on the block the time that a particular action took place).

Also, as companies' financial statements would be available in a common and universal ledger, this means auditing firms can start developing their own algorithms based on this ledger. This is the concept of **smart audit**. Each auditing firm can develop its own algorithm and launch this algorithm to audit a company (if the latter gave its agreement). These smart audits would detect frauds or discrepancies in the accounts according to the algorithm

In this paper, we'll go through most of the sections of financial audit by introducing the advantages of the Blockchain. We'll develop features brought by Request and its impact on audit. We'll introduce the concept of smart audit and look at the new issues of the Blockchain. Finally we'll present how to develop a model like Request at a commercial level.

The Request model

Request is a decentralized invoice which provides the support for the transformation of audit jobs. With this support, 'smart audits' would be possible thanks to an algorithm checking transactions in real-time on the Blockchain. Before going through what this system would allow in terms of audit improvement, here are the characteristics of Request.

An individual or a company can issue a Request.

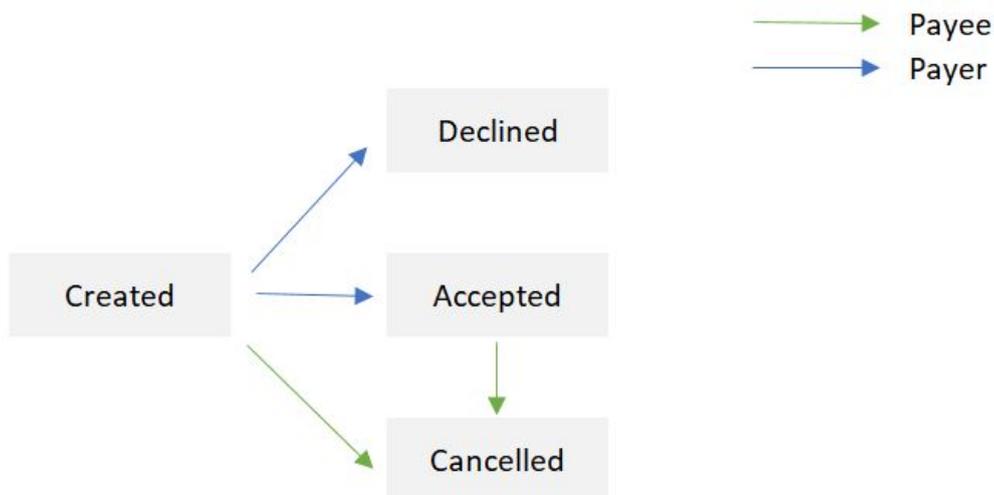
Request components (stored in the Blockchain):

Data associated with each Request are:

- the payer identity
- the payee identity
- the payment address
- the amount to be paid
- the currency
- the due date
- the Request status (created, accepted, declined, cancelled)

Also, invoice details will be stored off-chain in a decentralized way using Swarm and Filecoin technologies. Among other information, those details would include for example a budget code for accounting purposes, a VAT number for tax purposes.

Request status:



1/ Both the payer and the payee can issue a Request.

2/ When a Request is created (created), the payer accepts it (accepted) or denies it (declined).

3/ Only the payee can cancel a Request (cancelled). It can be cancelled when the amount paid is 0 (so either if the request has not been paid already or if the request was reimbursed).

4/ If the payee wants to cancel a request already paid, then he'll have to make a refund

5/ Only four possible statuses: open, declined, accepted, or cancelled

Request Ecosystem

List of third parties involved or impacted by the Request and the reasons why:

- **companies:** Request brings an instant payment solution for suppliers, it facilitates payments (no need to send the invoice and payment conditions by email or by post). It also brings the traceability of payment (payment-related billing). Request brings a simplified payment solution for customers (known payment information), a cheaper system, and a more secure system.
- **individuals:** they can do simplified payments (payment information known, no need to initiate a payment anymore but only to validate a Request), a secure payment (unshared banking information), a cheaper payment (no need to go through a third party like Paypal, Stripe, Bitpay that takes between 1% and 7% charge)
- **billing software:** payment management, interoperability (common database used).
- **accounting software:** simplicity (automation), immutability (common database), accuracy, authenticity (a decrease in human errors), traceability of invoices and payments (reconciliation), triple-entry system.
- **audit firms:** a better support for audit, elimination or improvement of certain tests.
- **States and Governments:** accuracy of tax declarations, reduction of fraud, money laundering, possibility of taxes collection in real time, transparency and legitimacy.
- **associations and NGOs:** transparency of their expenditure, legitimacy and trust.
- **citizens:** knowledge of government budgets and expenditures, transparency of institutions.
- **employees:** knowledge of financial health of a company.
- **IoT:** Request provides a framework for inter-machines payment.

Request main rules

Before going anywhere with the concept of smart audit in the Blockchain, it's important to get an overview of the main rules of a request (the overview is not exhaustive and some points are still to be defined).

It's important to understand what is a request and how it will work. It is a decentralized invoice stored in the blockchain and here are the different cases a request can experience:

- a cancellation of a request not yet paid
- a cancellation of a request already paid
- a refund
- a discount
- a late payment
- a partial payment
- an overpayment
- a mispayment

The amount of a request is fixed. For any cancellation, change or others, we'll use **subtracts** and **additional**s in order to manage the following cases:

In the case of a **cancellation of an Unpaid Request (or credit note)**: *the recipient/payee wants to cancel a Request that has been accepted but has not yet been paid by the payer.*

=> Cancellation of the Request

In the case of **cancellation of a Paid Request**: *the payee wants to cancel a Request that is already paid by the payer.*

=> either the request is refund then we can cancel the request

=> either the request is refund and then we do a subtract of the total amount of the request

This point is still to be defined

In the case of a **refund**: *the payer is unsatisfied of the realized service or delivered product (example: an Amazon delivery). The seller grants him a refund.*

=> the request is partially refunded.

In the case of an **overpayment**: *the paid amount exceeds the amount of the Request.*

=> if the payment is greater than the amount of the request, then the payment is denied.

=> however if the payer wants to pay more, he can add an additional 'tip' which increases the expected payment.

In the case of a **partial payment**: *the Request is partially paid.*

=> nothing happens. The Request remains in pending. There's no problem, a request can be linked to several different payments. We'll consider a paid request when the amount corresponds to the initial amount, plus the additional, less the subtracts.

In the case of a **late payment**: *the Request is still unpaid at its due date.*

=> Additional 'late fees / penalties'. ***To be defined***

In the case of a **mispayment**:

=> the request remains Accepted with no payment linked to it. as an unpaid request.

In the case of a **discount**: *price granted by the seller.*

=> this appears on the initial request

Additionals and **Subtract**s are always at the initiative of the payee and do not require a validation from the payer (except for the additional 'tip' which is at the initiative of the payer).

Customers and income

Customers and Income are two important sections of an audit. The main objective here is to check the completeness of account receivables, justify creditors, and to ensure the consistency of the profit and loss accounts by the correct allocation of sales.

The main advantages of Request in the audit of customers and income are:

- a **better allocation of trade receivables in accounting**. With the **payer ID** encrypted in the request, each request is identified and allocated to a customer code and goes to a customer account.
- no more **reconciliation between the invoicing and accounting system**. With the triple-entry accounting function, the ledger is shared and immutable.
- no more reconciliation of banks with the opened invoices. The **payment is linked to the request**. The accountant has instant knowledge of the company's pending invoices.
- a **better cash management** for several reasons: a correct and current information of the pending requests, a faster payments collection (no delays between the time the payer initiates the payment and the time the payee receives it) and a better company's DSO (days sales outstanding) as the reputation system gives a good reason to the customers to pay their invoices on time.
- a **better management of the provisions for bad debts**. The accountant or auditor can easily detect the doubtful customers thanks to a reputation system for bad payers.
- a **partial automation of invoices verification**: with Request, we identify the **payer ID** and the **payee ID**. There's no risk of fraudulent invoices as the system records the identities of both parties. We **outperform the use of sampling** as we don't check a sample of invoices anymore but all the requests automatically.
- a **facilitated cut-off test**: with the period of the sale recorded in the request, we can easily detect the provisions to be booked. This results in a consistent profit and loss statement.

See the details of the audit of trade receivables and income in the tables below:

Audit of account receivables:

| |
|---|
| 1/ Reconciliation of subledger accounts with accounting |
| Subledger account = the outstanding balance of trade receivables (by customer) Trial balance = the outstanding balance of the account receivables (total receivables) Ageing list = list of the pending invoices (by invoice) |
| Objective = validate the correct allocation of trade receivables in the accounting. |
| Request allows the automation of this reconciliation test. How? By the automatic allocation of the Request to a customer code and a customer account. |
| 2/ DSO analysis (days sales outstanding) |
| It represents the average payment period for customers (in number of days) DSO = (total trade receivables / total cost of sales) x 360 |
| Objective = internal management of a company's cash flow |
| Request does not contribute to the calculation of the DSO. However it improves a company's DSO by giving a reason to the customers to pay their invoices on time (with a reputation |

system).

3/ Analysis of 'Invoices to issue' provisions

Provision 'invoice to issue' = when a sale corresponds to an earlier period than the date of the invoice.

Example: a company issue a customer invoice in february whereas the service has been done in January. We need to book a provision in order to allocate this sale in January and remove it from February. In order to have a P&L with a correct allocation of sales.

With the period of the sale entered in the code of the Request, it is easy to detect the invoices to be provisioned. This would facilitate their completeness.

How? With the period of the prestation entered in the Request.

4/ Transactions test

With a sample of invoices, we check:

1. if the service or good was realized or delivered
2. if the invoice concerns the audited company
3. if the invoice is booked in the correct account
4. if the VAT is calculated correctly

Objective = check the completeness of the trade receivables and their correct bookkeeping.

Request solves this test for the point 2 (each request is attributed to a company) and the point 3 (each request has a budget code for accounting purposes). Furthermore, Request removes the use of sampling method as the test is applied to all invoices.

5/ Provisions for bad debts

Objective = look at the client recovery to detect the outstanding invoices and depreciate doubtful receivables.

Thanks to a reputation system of bad payers, Request allows the company to better manage its doubtful customers. The depreciation of its receivables is facilitated.

6/ Cut-Off test

Select a sample of invoices whose amount is above a predefined threshold in order to verify that the invoice is recorded on the correct fiscal year.

If the invoice or part of the invoice corresponds to the following fiscal year, then a provision for 'Deferred Product' must be registered.

Objective = ensure the consistency and accuracy of the Profit and Loss Statement.

As the period of the sale is included in the Request, the detection of invoices requiring a cut-off provision is easier. And not just for a sampling of invoices but for all invoices.

7/ Analysis of 'Deferred Products' provisions

| |
|---|
| <p>Provision for Deferred product = when a product corresponds to the following period. Example: a company issue a sale invoice in advance. Vérifier l'exhaustivité des PCA - use of judgement</p> |
| <p>Objective = check the completeness of the Deferred Products provisions - Use of judgement</p> |
| <p>Request does not contribute to the analysis of these provisions. However, as the period of the product is entered in the code of the Request, it is easy to detect the invoices to be provisioned. This would facilitate their completeness. How? with the period of the prestation entered in the Request.</p> |
| <p><u>10/ Intercompany accounts reconciliation</u></p> |
| <p>Objective = reconcile the accounts of a company's subsidiaries. See 'intercompany transactions' part</p> |
| <p>Thanks to the interoperability of the Blockchain, Request builds a shared ledger that eliminates the reconciliation of intercompany accounts. How? An invoice isn't anymore recorded in two separate independent systems but shared in a decentralized ledger (see more information in the intercompany part of the yellow paper)</p> |

To this can be added the **revaluation of account receivables in foreign currency** at the end of the fiscal year, the **analysis of creditors** customers and the **analytical review** of sales. Request does not contribute much here. Auditors will be able to spend more time on this.

Audit of Income:

| |
|---|
| <p><u>1/ Reconciliation of the sales ledger with the billing system</u></p> |
| <p>Check that all sales invoices are recorded in accounting (=> reconciliation billing system with accounting system).</p> |
| <p>Objective = validate the consistency of the sales.</p> |
| <p>With Request and the triple-entry accounting system, all sales invoices are recorded in the shared ledger - immutable</p> |
| <p><u>2/ Test on sales recurrence and periodicity</u></p> |
| <p>Objective = validate the consistency of charges by their recurrence and periodicity. Example: consulting services invoiced each month of a similar amount is a consistent sale.</p> |
| <p>Request does not contribute to this test but it has the potential to allow a better analysis.</p> |
| <p><u>3/ Classification of invoices according to their tax code (inland, intracom, export)</u></p> |
| <p>Objective = validate the correct amount and allocation of collected VAT.</p> |

| |
|---|
| Request does not contribute to this test. |
| |
| 4/ Check of exchange rate used for sales invoices in foreign currency |
| Good use of exchange rate. |
| Request does not contribute to this test. However it could force companies in the same industry to use the same method. |

Suppliers and charges

The main objective here is to check the completeness of account payables, justify debtors, but above all to ensure the consistency of the profit and loss accounts by the correct allocation of purchases.

The main advantages of Request in the audit of suppliers and charges are:

- a **better allocation of trade payables in accounting**. With the *payee ID* encrypted in the request, each request is identified and allocated to a supplier code and goes to a supplier account.
- a **partial automation of invoices verification**: with Request, we identify the *payer ID* and the *payee ID*. There's no risk of fraudulent invoices as the system records the identities of both parties. We **outperform the use of sampling** as we don't check a sample of invoices anymore but all the requests automatically.
- an **improvement in the audit of charges**: with a *budget code* recorded in the request, each request is allocated to a specific purchase account. With the Blockchain, we can have an universal database of suppliers with attributed budget codes. A smart audit would detect wrong allocation of purchases in a company's financial statement. **For example, a request with the supplier Nespresso would only be recorded with a budget code related to the purchase of coffee.**
- no more reconciliation of banks with the suppliers opened invoices. The **payment is linked to the request**. The accountant has instant knowledge of the company's invoices to be paid.
- a **facilitated cut-off test**: with the period of the sale recorded in the request, we can easily detect the provisions to be booked. This results in a consistent profit and loss statement.
- a better **detection of old debts**: if an unpaid supplier invoice is older than 3 years old for example, a smart audit can flag the request as an invoice to be reversed.

See the details of the audit of trade payables and charges below.

Audit of account payables:

| |
|--|
| 1/ Reconciliation of subledger accounts with accounting |
|--|

| |
|--|
| <p>Subledger account = the outstanding balance of trade payables (by supplier) Trial balance = the outstanding balance of the account payables (total suppliers debts) Ageing list = list of the pending invoices (by invoice)</p> |
| <p>Objective = validate the correct allocation of trade payables in the accounting.</p> |
| <p>Request allows the removal of this reconciliation test. How? By the automatic allocation of the Request to a supplier code and a supplier account.</p> |
| <p><u>2/ DPO analysis (days payable outstanding)</u></p> |
| <p>It represents the average supplier payment period (in number of days) $DPO = (\text{total of trade payables} / \text{total cost of goods}) \times 360$</p> |
| <p>Objective = internal management of a company's cash flow</p> |
| <p>Request does not contribute to the calculation of the DPO. A reputation system would be an incentive to pay suppliers invoices on time.</p> |
| <p><u>3/ Analysis of 'Unreceived Invoices' provisions</u></p> |
| <p>Provision for Unreceived Invoices = when a charge corresponds to an earlier period than the date of the invoice. Example: the electricity bill for the month of december is received in January, the invoice is dated on the 3rd of January. We need to book a provision in order to allocate this charge in december and remove it in January. If not, we have a wrong P&L because charges aren't allocated on the corresponding periods.</p> |
| <p>Objective = check the completeness of the Unreceived Invoices provisions - Use of judgement</p> |
| <p>With the period of the sale entered in the code of the Request, it is easy to detect the invoices to be provisioned. This would facilitate their completeness. How? With the period of the prestation entered in the Request.</p> |
| <p><u>4/ Transactions test</u></p> |
| <p>With a sample of invoices, we check:</p> <ol style="list-style-type: none"> 1. if the service or good was realized or delivered 2. if the invoice concerns the audited company 3. if the invoice is booked in the correct account 4. if the amount of the invoice corresponds to the order form 5. if the VAT is calculated correctly |
| <p>Objective = check the completeness of the trade payables and their correct bookkeeping.</p> |
| <p>Request solves this test for the point 2 (each request is attributed to a company) and the point 3 (each request has a budget code for accounting purposes). Furthermore, Request removes the use of sampling method as the test is applied to all invoices.</p> |

| |
|--|
| <u>5/ Seniority of suppliers debts</u> |
| Objective = a supplier debt must not remain pending for payment more than a certain number of years. |
| A smart audit can detect the old requests unpaid and flag them as invoices to be reversed. |
| <u>6/ Cut-Off test</u> |
| Select a sample of invoices whose amount is above a predefined threshold in order to verify that the invoice is recorded on the correct fiscal year. If the invoice or part of the invoice corresponds to the following fiscal year, then a provision for 'Deferred Charge' must be registered. |
| Objective = ensure the consistency and accuracy of the Profit and Loss Statement. |
| As the period of the charge is included in the Request, the detection of invoices requiring a cut-off provision is easier. And not just for a sampling of invoices but for all invoices. |
| <u>8/ Analysis of 'Deferred Charges' provisions</u> |
| Provision for Deferred Charge = when a charge corresponds to the following period. Example: a car's insurance premium for the year 2018 is invoiced in december 2017. We don't want the charge to appear in the books in 2017 but in 2018. So we book a provision to remove the charge from 2017 and input it in 2018 (while the invoice is still recorded in december 2017 of course). Vérifier l'exhaustivité des CCA - use of judgement |
| Objective = check the completeness of the Deferred Charges provisions - Use of judgement |
| With the period of the sale entered in the code of the Request, it is easy to detect the invoices to be provisioned. This would facilitate their completeness. How? With the period of the prestation entered in the Request. |
| <u>10/ Intercompany accounts reconciliation</u> |
| Objective = reconcile the accounts of a company's subsidiaries. cf 'intercompany transactions' part |
| Thanks to the interoperability of the Blockchain, Request builds a shared ledger that eliminates the reconciliation of intercompany accounts. How? An invoice isn't anymore recorded in two separate independent systems but shared in a decentralized ledger (see more information in the intercompany part of the yellow paper) |
| <u>11/ Revaluation of trade payables in foreign currency</u> |
| Objective = calculate the revaluation of suppliers debts at the end of the fiscal year. |
| Request does not contribute to this test. |

| |
|---|
| 12/ Analytical review |
| Objective = internal review of charges for a better management. |
| Request does not contribute to this test. |

Audit of charges:

After the audit of suppliers debts, we have the audit of the expenses. The objective is to validate the consistency and completeness of the income statement. See the details below:

| |
|--|
| 1/ Test on entries description |
| Objective = analyse the consistency of charges by their description on the ledger. |
| Request eliminates this test with the budget codes verification. |
| |
| 2/ Test on expenses recurrence and periodicity |
| Objective = validate the consistency of charges by their recurrence and periodicity. Example: a rent of the same amount each month is a consistent charge. |
| Request does not contribute to this test. |
| |
| 3/ Test on invoices sampling |
| Objective = select a sample of invoices and check the amount, the currency, the VAT. |
| Request prevents from recording a wrong amount or currency in the accounting ledger. An invoice isn't anymore just a piece of paper sent between two companies but is a entry written on the ledger. |

To go further, one can imagine seeing the emergence of **comparative and statistical audits**. Indeed, thanks to the universal ledger we can compare the financial statements of different companies. For example we could compare the expenses of several companies in the same sector of activity, and more or less similar size. Significant differences in the nature of the expenses or in the volume should alert the auditor. For example we can compare the payroll to revenues of two IT consulting companies. This gives a ratio that the auditor can assess as a good or bad ratio of wages. These techniques are **smart management controls**.

Intercompany transactions

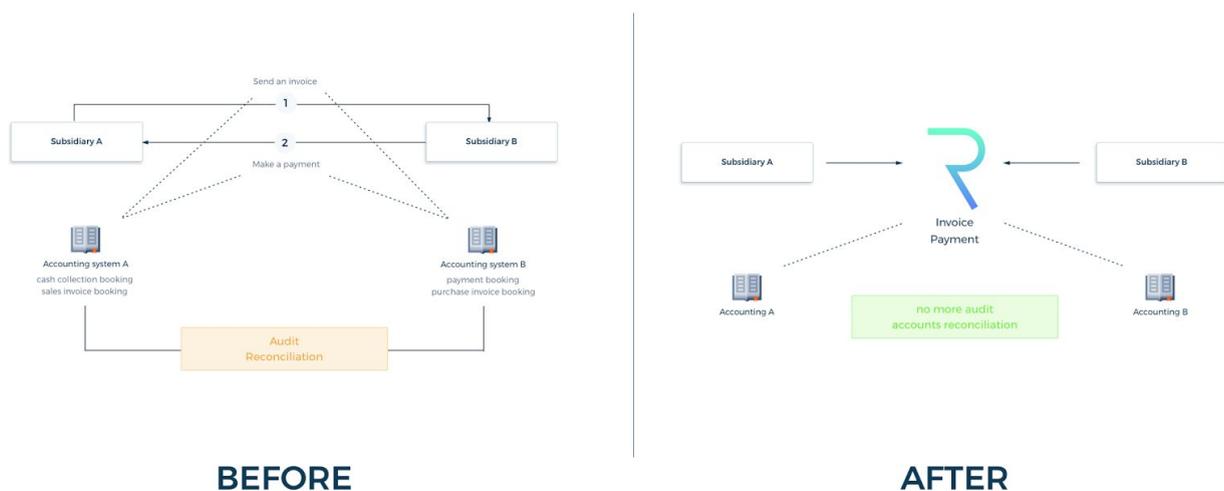
One of the biggest impact of the Blockchain in accounting: the **management of intercompany transactions** and financial consolidation of an international group. When we think of transactions between subsidiaries, we immediately think of 'consolidation process' by eliminating them at the end of the fiscal year. Yet, before the consolidation, these transactions are an integral part of the accounting closing through the intercompany reconciliation process.

Intercompany transactions represent a complex management. With a large volume of internal transactions, it requires a good number of players within the group and different accounts. Their existence comes from varied situations. Indeed, apart from equity accounts, equity investments, fixed assets and inventories, all other accounts may be subject to intra-group flows. Between the sale of goods and services, cost sharing, cost allocations, and royalties, intra-group transactions are estimated to be more than 30% of international business flows.

The reconciliation of these inter-subsidiaries accounts is a tedious job every year for financial teams. The main reasons for these intercompany differences are related to:

- different closing dates
- shifts in recording dates (the "cut-off")
- the conversion of transactions denominated in foreign currencies
- the capitalization of a purchase by the acquiring company
- the acquisition of VAT by the acquiring company

Below the **Intercompany reconciliation process** today and the reconciliation process with Request:



Currently, when a subsidiary A sends an invoice to a subsidiary B, the two companies record the invoice in two independent accounting systems. Company A registers a sale in its system linked to third party B. And company B records a purchase linked to third party A. As the same invoice is registered independently in two different systems, nothing prevents from the omission of the invoice by one of the two subsidiaries.

Thanks to a digital invoice such as Request, the registration of an invoice could not be omitted. And the reconciliation of inter-subsidiary accounts is largely facilitated by the decentralized nature of the Blockchain.

This is the most classic case. To this we can add the problems of **reconciliation of the provisions** related to an invoice. If an invoice is a benefit concerning the following fiscal year, the subsidiary A

will book a provision for 'Deferred Product' while the subsidiary B will book a provision for 'Deferred charge' (prepaid expenses). These "cut-off" provisions cause a significant number of **intragroup discrepancies**.

A system like Request would easily identify these discrepancies related to the "cut-off" by writing in the code of the request the period of the service. With this, the auditor can easily identify expenses and incomes that should be cut-off.

Taxes and duties

Can we use Blockchain to collect taxes? Yes, and governments are well aware of this. A system of digital invoices like Request will allow two things.

First, a system like Request would allow to **declare VAT instantaneously**. Currently, each company declares its VAT annually or quarterly. It calculates all its incoming and outgoing invoices and then makes its declaration. The VAT return is almost always a nightmare for the accounting teams of companies. Between VAT credits, reversals, past mistakes to be corrected, and changes from previous quarters, the VAT return is a painful task for the company. With the Blockchain, when an invoice is issued and validated by third parties, the VAT could be declared to the State in real-time. In addition, the interconnection of the Blockchain will allow to recover VAT as soon as an invoice is cancelled for example. There is no need to make a corrective statement that comes to cancel and replace the latest one. The VAT collection service knows in real time the amount of VAT due by a company.

Second, a system like Request allows not only the VAT declaration in real time but also **the payment in real time**. Indeed, as the declaration is made simultaneously, the payment could be the same.

We can go further because we can assume that a company will no longer have to declare its VAT because the company doesn't collect the VAT anymore.



Use case: a company A sends a sale invoice to a company B for an amount of 100 € (tax excluded). A 20% value added tax applies. Currently, the company B pays 120 €, VAT included. At the end of the year, A declares its taxes to the State and pays the 20 € of VAT collected on this invoice. With Request, when A sends an invoice of 100 € with a VAT of 20%, B would have to pay 100 € to the company A and can pay directly the 20 € of VAT to the State. Instantly. This makes the collection of taxes by the government fluid, instant and mistakes-free.

Then we can raise cash issues for company A which no longer collects VAT during the year. When previously it collects 20% of the total of all its sales invoices before paying them back to the State at the end of the year, the company does no longer receive the VAT money on these sale invoices.

To compensate for this, Request could make available the non-mandatory option of paying taxes directly to the State. If the company chooses not to pay its taxes directly, then the calculation of VAT is still done in real time (which represents a great improvement over the current system). If the company chooses to pay its taxes instantly to the State, then we can imagine that the government offers the company a decrease of the VAT rates. Indeed the government has a shortfall of several millions of euros each year because of false declarations. There are not enough tax auditors to control all companies and it costs time and money. If tomorrow the government has a good picture of companies taxes and if the VAT is paid simultaneously, then what prevents it from lowering the VAT rates?

Credit analysis

Credit scoring consists of assigning a rating to the credit applicant based on its capacity to repay (whether an individual or a company). It allows banks to lend money by evaluating the risk of credit default via a statistical score. Banks, major credit rating agencies (such as Moody's, Standard & Poor's, Fitch), credit insurers (such as Euler Hermes, Coface) each have their own method of calculating a company's credit risk. Each actor has its own notation method. However, the calculation is not easy because of two factors. First, it is common for a company's financial statements to be published 6 months after their fiscal close, especially for small businesses. Second, information is often limited to assess the financial health of a business. This results in a company rating made or updated only belatedly with more than 6 months lag. To resume, credit analysis is a painful task because of **non-transparent and outdated financials**.

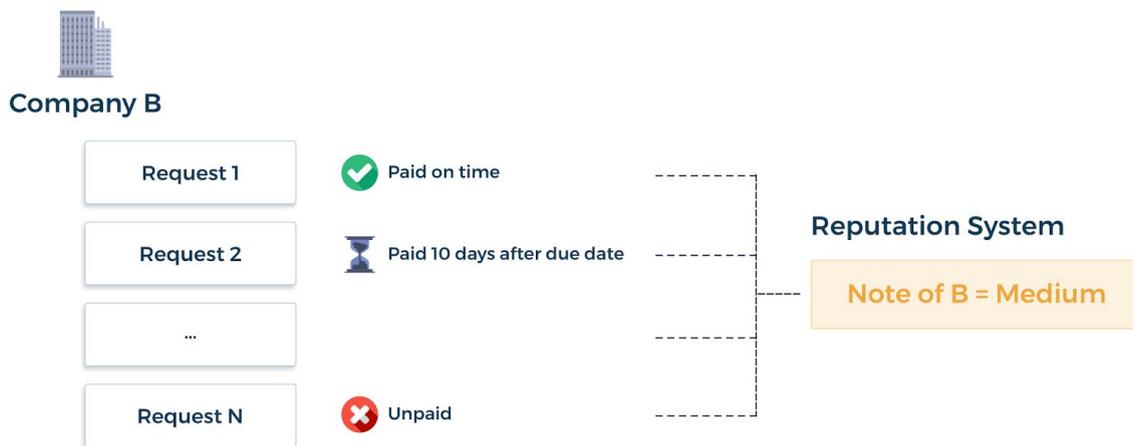
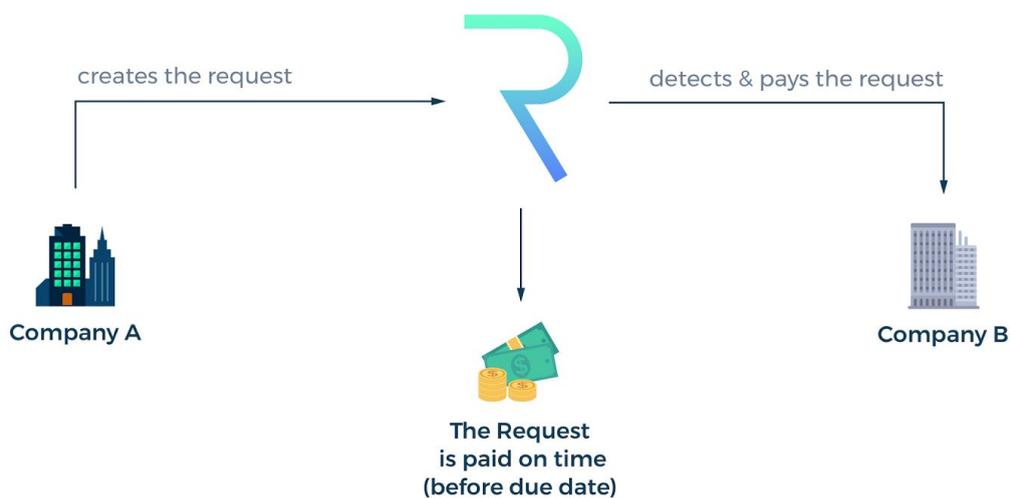
A company can improve its scoring by communicating **correct and current information** about its financial situation. This is what Request allows. With Request Network, access to a company's financial data and invoices is available on a decentralized database. This shared database not only enables instant invoice auditing and therefore improves the company's credit scoring. It also makes the pending invoices of a business available in the credit market (available for factoring in the retail market).

With the reputation system of Request Network, it is possible for companies (and individuals) to improve their reputation as 'good payers' by paying their bills on time. This is possible thanks to the **visibility** and **transparency** allowed by the Blockchain. In addition, the pending invoices and the financial situation of a company is visible instantly thanks to the Blockchain. This allows the credit applicant to get an almost instant rating. Request allows to assign a score to a company or an individual thanks to a system of reputation updated as regularly as possible.

How does the Request reputation system work?

Request takes into account several criteria:

- the effective payment
- the timing of payment (on time or late)
- the number of requests issued
- the status of the requests (accepted or declined)



If a company's rating is bad, then the company waiting for the payment will be able to easily anticipate and book a provision for doubtful debt. The reputation system should enable a company's financial teams and auditors to better anticipate the **depreciations for bad debts**.

How to manage the confidentiality issue?

In a visible and transparent system allowed by the Blockchain, it is necessary to be able to manage confidentiality issues. Indeed, companies won't make their invoices public to customers and suppliers. To do this, we will have to rely on the reputation system. The company would have to give the Request Network the right to access its invoices in order to launch an algorithm calculating its reputation.

Because of instant invoice auditing and credit scoring updated on a regular basis, companies have easier access to factoring, credit or other financing. Invoices via Request become a **source of shared**

liquidity for invoicing financing and factoring. Small businesses will be able to access short-term financing by trading their receivables.

The new issues raised by the Blockchain

We're still in the early days of understanding what Blockchains can do. But the long-term potential of the technology is huge. Especially in the finance area with a significant new possibilities. That says new possibilities also says new issues. Here are the challenges to be faced in the next few years:

- the **security** issues => set up security for possible theft of signature keys for example
- the **cost** of implementation
- the **privacy** issues => manage data confidentiality and privacy
- a lack of **agreed standards**
- a limited **scalability** of the Blockchain today

Commercial development: onboarding and transition

Accounting practices in any firm are slow to adapt and often unwilling to switch of system. Indeed it is far from easy to switch of accounting system because it requires to upload all the historical data in a new system. When it concerns an international group with several subsidiaries in different countries, it takes years to switch of system.

The aim of Request isn't to replace SAP or any accounting system but to **provide the source of accounting entries** for both invoices and payments. To develop Request, we'll go first to the SMBs and startups such as Quickbooks, Xero, Sage, Wineur. Then larger softwares (like SAP) will be able to plug it to the system thanks to Request Network interoperability.

On one hand, Request makes it easier to create and share an invoice. On the other hand, it makes easier to pay it on time. This results for companies in getting paid on time because the payers don't want to get a bad reputation on the market. Such a system might take time to implement but this is definitely the direction we are taking thanks to the Blockchain.

If you have any question or input on the impact of Blockchain on accounting and auditing practices or if the subject matters to you, come talk to us on our [Slack!](#)