**Exploring possible applications of micropayments in IoT applications and using the Bitcoin blockchain and the Lightning Network to do so**

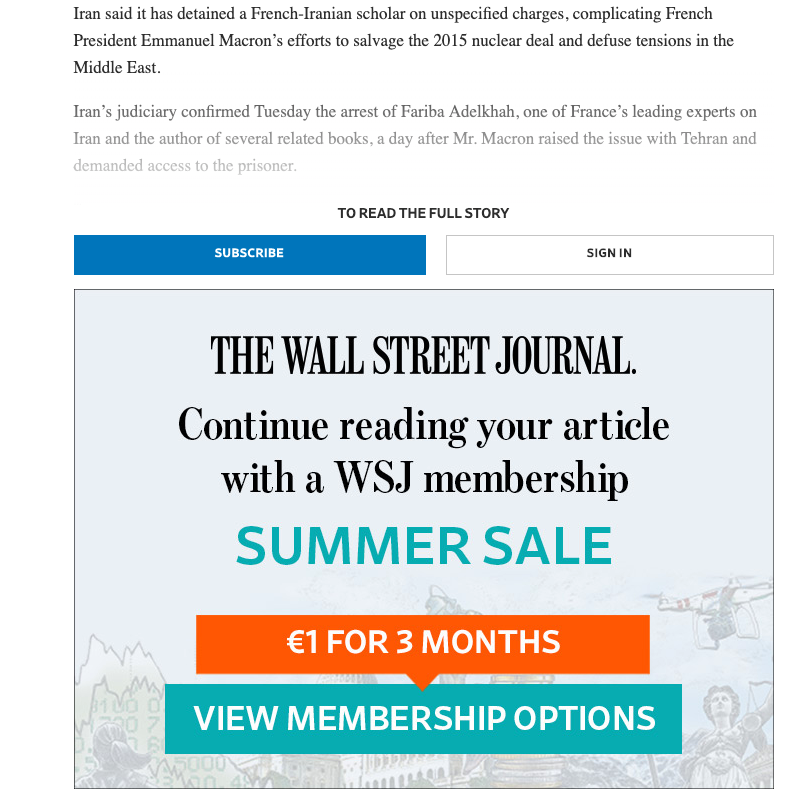
My current plan of action for my report:

1. Discuss the current methods used for payments (for any payments based on fiat currency whether it be in cash or online). These problems will include transaction fees, trusting 3rd party banks, other security issues such as entering credit card details into online websites in order to purchase goods & services. What will also be discussed is the way that many goods and services are currently being billed: bulk payments, flat-rate payments and subscriptions. Also discuss the current way that online content creators are making money: either by showing adverts or by putting up pay-walls. What can also be discussed is how difficult it is for some people in SA to open bank accounts which then forces them to be paid in cash which in turn makes them vulnerable to theft.
2. What is Bitcoin and what is the Lightning Network and how does it work (Will go in depth here about exactly how Lightning Network payments work using HTLCs (Hash Time Lock Contracts)). Then, how can the Bitcoin and Lightning network be used to address and solve all the previously mentioned problems (security, fees, trust) and how it can be used to make microtransactions a realisable reality.
3. Further discussion on microtransactions and explorations into the different areas it can be used to change the way money flow and data flow currently works. This will make up the majority of the project. I will explore different avenues in which micropayments could be useful and then critically evaluate how these integrations would work and whether or not the results would be desirable and beneficial (ie: where will it be useful and where will it just be an unnecessary hassle). I could then implement a few examples as proof-of-concept and for demonstration purposes for some of the different areas.

Some of the areas I have considered exploring so far are as follows:

* Pay-as-you-consume content:

Many online content creators often opt for selling advert space or putting up pay walls such as the following in order to make an income for their content:



These kinds of methods often deter users from websites and they will often opt not to read an article, watch a video or listen to a podcast because they do not want to commit to a monthly subscription when all they want is 1 piece of media. Nor do they want to have to enter their credit card details into an online site in order to buy the piece. The result is that users do not consume any content and content creators do not get paid. (An article on the pros and cons of pay-walls: <https://www.mobiloud.com/blog/paywall/>)

It would be interesting to explore the idea of being able to charge a user in micro-transactions for the exact amount of content that they consume and to do this in real-time.

If I implemented this I could explore the following:

* Creating and processing microtransactions as a user scrolls down the page of an article so that they pay for what they read.
* Attaching microtransactions to every second of an online video so that a user is charged, in real-time, for the exact number of seconds of the video that they watch.
* Similar to the above for charging for things like music and podcasts.

All of the above would involve figuring out how to attach value to individual pieces of data and charging the user accordingly (and not charging them again when they consume the same piece of data twice).

If I figure out how to do that then I could extend this and explore the idea of attaching value to other types of data. For example: can Facebook and Google possible pay users in microtransactions and in real-time for using their data (location etc)?

* Pay-as-you-go resources: (IoT applications. I am keen to implement something like this)

Pay-as-you go while using resources controlled by an IoT device. This could be a water tap, electricity etc.

* Pay-as-you-go services:
* This would be like the ‘Lightning Cab’ video. A person gets in a taxi or Uber and is charged incrementally in microtransactions and in real time based on the duration and distance they have traveled.
* Microtransactions used to incentivise activities:

With microtransactions together with the Bitcoin smart contracting language, it is possible to attach tips to things in order to incentivise people to complete certain tasks. For example: I could write an email to a celebrity (or anyone that I think will be in high demand) and I could attach a small (or large) tip to the email and using the smart contracting language, I could ensure that if the celebrity responds within a certain amount of time then they will be paid that tip, otherwise it will be paid back to me. This incentivises the celebrity to respond to my email in a timely manner.

* Using Microtransactions and pay-to-script addresses to raise funds for charities:

Currently, if you want to donate money to a charity via an online website you would only really be able to make large payments to the charity (because the fees attached to donating small amount would be too high). You would also need to trust that the money is going to the right hands. With micropayments it will be possible to anyone to make tiny payments to these charities with any left over change that they might have and they could do this without having to pay large transaction fees. Because it would be so easy to donate money, it could result in many more people contributing and thus a significant amount of money being raised for charities.

* Using micropayments to avoid spam:

By being able to requiring small fees to be attached to an action such as voting online for the content that a content creator next, it is possible to minimize spam and any bot systems.

Or even attaching small fees for sending an email so that sending spam mail becomes costly.

* Using micropayments to tip car guards:

Many people are starting to carry less small change due to the ease of paying with debit/credit cards. As a result, people are often unable to tip parking guards (a popular job in South Africa). One solution would be to enable parking guards to receive payments through an application such as Snap-Scan in which case the parking guards would carry a QR-code that customers could scan and pay too. However, this solution would require parking guards to have a bank account which would also require them to have some form of identification and proof of residence, both of which they may not have. This solution also would still use 3rd party banks and so small transactions would still have high fees attached.

It would be interesting to investigate whether the Lightning Network could be used to solve this problem because no form of identification etc is required for a parking guard to have a Lightning address and also the small microtransactions that customers would make would not result in large fees. The difficulty is that currently for person A to pay person B with Lightning, person B would need to generate a new invoice for person A each time (ie: a fixed QR code would be a problem). Investigations could thus be made regarding whether or not, given the maths and implementation of the Lighting Network, a solution could be formed for this problem.

User Requirements

I will be focusing on the following 2 users:

* Customers wishing to make online purchases of goods and services.
* Service or Goods providers who wish to sell their goods and services online.

|  |  |
| --- | --- |
| **Label** | **User Requirement Description** |
| UR1 | Customers require a way to control their own money without having to trust a custodial 3rd party. This means that users require a way of making online purchases without having to give out private information such as credit card details. (In terms of Bitcoin, this means that users should not have to distribute and trust any 3rd parties with their private keys in order to purchase goods and services.) |
| UR2 | Customers require a way to make online purchases without having to wait an unreasonable amount of time for the transactions to be confirmed. These payments should happen almost instantaneously. |
| UR3 | Customers require a way to make small online payments without having to pay large transaction fees. |
| UR4 | Customers require a way to pay, in real time, for the exact amount of good or service that they are using instead of being forced to commit to bulk payments of goods or services. |
| UR5 | Customers require that the payments be easy to make and easy to understand. |
| UR6 | Servers require a way to allow customers to pay by making ‘continuous’ microtransactions in real-time so that they do not need to put up pay-walls or sell space to advertisers. |

Functional Requirements:

|  |  |  |
| --- | --- | --- |
| **Label** | **UR** | **Functional Requirement Description** |
| FR1 | UR1 | When a customer wants to make a purchase of some sort, an invoice should be made available to them which they can then pay and in which they can see the details of the payment. The transaction will be broadcasted on the Bitcoin Lightning Network and the good/server provider should be able to see that the invoice has been paid by the specific customer. The payment thus happens completely on the customers side without them having to distribute private information and without having to use a 3rd party such as a bank to control the payment. |
| FR2 | UR2 | After paying an invoice, the customer should receive the good/service that they have paid for in the shortest possible amount of time. This means that after having paid the invoice, the transaction should become verified and trusted immediately. |
| FR3 | UR3 | When users are making payments, there should be no restriction on the amount of the payment and the value of the payment should not affect the fees or the time it takes for the transaction to be considered valid. |
| FR4 | UR4 | Customers should be able to pay for the content they use as they use it. When they start viewing the content, their Bitcoin balance should be updated in real-time so that they can see their balance change as they use the content. When they stop viewing the content, their balance should stop changing. |
| FR5 | UR5 | The user should only require a Bitcoin wallet and should not be required to set up a complex Lightning wallet or set up their own Lightning network channels. All the complexities should happen behind the scenes and all that the user should be required to do is indicate what they would like to buy and that they agree to use their Bitcoin to do so. |
| FR6 | UR6 | It must be possible to connect a certain small, discrete pieces of data so that invoice can be created for individual pieces of content so that customers can be charged accordingly. |
| FR7 | UR6 | Every paid invoice should be recorded so that a user is never charged twice when they re-consume previously consumed content. |