

Gib: Streaming Micropayment Gift Cards

Authors: Adelya Latifulina & Alexander Mozeak

Research funded by Grant for the Web, sponsored by Coil, Mozilla, and Creative Commons

[adelya, alex]@astar.app

New York, NY

***Abstract*--Gib is a browser tool that allows users to generate and share premium Internet content through gift cards, which stream micropayments over the Interledger Protocol. The advantage of this experience is three-fold. For content consumers, it allows greater access to premium content while maintaining their anonymity. For content creators, it allows reach to a greater audience without defaulting to a monetization model of ads and/or subscriptions, yet it can still be used in conjunction with both. Lastly, for generous folks who wish to share access to exclusive digital content, Gib enables their intention with a few clicks.**

***Keywords*--web monetization, Interledger Protocol, micropayments, gift cards.**

I. INTRODUCTION

Sharing paid Internet content is often difficult. The sender and receiver must have paid accounts with the content provider and, in the case of crypto-monetized content, both must be technologically savvy as well as on the same blockchain. Content creators lose out on a potential audience that may never experience the value of their content. Low-income and unbanked users are often left out of a walled garden of premium content.

To solve this problem, we set out to create Gib; a tool that helps people share web monetized content just as easily as they would normally share hyperlinks. We underwent a rigorous process of researching the Web Monetization specification and associated protocols, generating our own specifications for various form factors, and problem-solving around the challenges that arose from various approaches. Ultimately, our research brought us closer to the seamless end-user flow that we initially envisioned, for all participants in the ecosystem.

This paper walks through that research and development process. In Section 2, we provide a survey of the relevant technologies and implementation details that we considered for Gib. Section 3 details the user flows and challenges that we encountered when designing the architecture to enable different form factors, lays out the findings of our research, and describes the resultant user flow. In Section 4, we summarize the process and provide the conclusion.

II. TECHNOLOGY SURVEY AND CONSIDERATIONS

A. The Web Monetization API and the Interledger Protocol

Gib is based on the Web Monetization API. This proposed specification allows anonymous streaming of payments from a client to a web host¹. In order to enable the Web Monetization API, we need to implement the **monetization** Payment Method Identifier² and the Payment Handler interface³.

Web Monetization itself is based on the Interledger Protocol (h.r.a. ILP), which is a method that allows payments to be made between any two ledgers, regardless of which blockchain each ledger is on⁴. ILP is set up to run over HTTP with HTTPS accessible endpoints, which allows it to utilize existing industry tools and services “for hosting, load balancing, Distributed Denial of Service (DDoS) protection, and monitoring”⁵.

ILP is significant because previously, if parties wanted to transfer money online they had to do it one of two ways. The first way would be to have an institution like a bank, or a financial company connect one account to another. The second way would be to transfer payment over a blockchain, whereby the receiving account must be on the same chain. The Interledger Protocol provides a third way; using encrypted escrow accounts to establish a connection across payment ledgers, all the while maintaining security and stakeholder trust.

B. Gift Cards

To encourage sharing premium content on the Internet, we used the real-world model of gift cards. When considering gift card types, there are typically two: closed-loop and open-loop⁶. Closed-loop gift cards are tied to a specific vendor, while open-loop gift cards can be used across vendors. Because the closed-loop approach would require direct integration with the given vendor's website, Gib is modeled after the open-loop gift card.

C. Ease of Use

To find the best form factor for Gib, we focused on what would provide the lowest friction for the end-user, while still respecting their security and privacy. Between two alternatives - a web component and a browser extension - we considered four metrics; ease of set up, payment availability, browser compatibility, and anonymity (*Table A*).

To aid in our product thinking, we created three personas: Alice, Charlie, and Bob. Alice is a premium content consumer who wishes to share web monetized content. Charlie is a content producer who wishes to deliver quality content and grow their audience. Bob is a content consumer who doesn't yet have access to premium Internet content, but is the recipient of Alice's sharing.

III. IMPLEMENTATIONS AND CHALLENGES

A. Link Generation/Sending Challenges

In the first stage of the user flow, Alice should be able to pay for an increment of content access, generate a link, and send it to Bob. However, Web Monetization is built around streaming payments, rather than one-time purchases. In order for Gib to work within the current specification, it would have to either accept a one-time payment from Alice and store those funds to be streamed later, or be given access to Alice's Web Monetization-enabled wallet and allowed to stream the payments from there as Bob browses.

Considering the former case, where Alice's one-time payment is stored for later consumption by Charlie's page, one potential design could have Gib set up as a payment streaming server. In this scenario, Gib would manage an escrow account where Alice's funds are held. It would then listen for a signal indicating that Bob has accessed the content and

begin streaming payments to Charlie accordingly. The challenge of this approach is that Gib would have to act as a centralized processor of payments. Implementing this would likely entail additional regulatory overhead, as Gib would have a fiduciary responsibility to manage users' funds and be liable for any mishandled funds.

In the latter case, where Gib is given permission to stream payments on Alice's behalf, the challenge would be in setting up access into Alice's payment method and opening the stream from that access point when Bob visits Charlie's site.

B. Link Opening Challenges

When Bob clicks a link that has been sent by Alice, he is directed to Charlie's page and given the same level of access that Alice had while browsing with monetization enabled. One way this could be achieved for an open-loop card is by first routing Bob's browser through a Gib proxy site, where a Service Worker implementing the Payment Handler API^{3,7}, as well as the `monetization` Payment Method Identifier, would be installed, then redirecting Bob to Charlie's site. This would require some overhead. It would also be challenging to tell when Bob has left Charlie's site, though we could set up a ping system to relay to the server when the content is no longer being accessed.

Conversely, if we were to take a closed card approach, Gib would be a web component⁸ - installable to Charlie's website, would listen for incoming Gib visitors, and then run the process of setting up the service worker on Bob's browser and calling out to the funding source to start streaming payment.

C. Implementation Summary

Weighing out the choice between web component and browser extension, our research indicates that an extension would lead to a smoother user experience. The assumption is that Alice, the user who wants to share the gift card, and Bob, the recipient of the gift card, would both install the Gib browser extension. This will give us the ability to fulfill all of the requirements necessary for web monetization to run on Bob's browser.

However, the main drawback of this approach is that Bob, the user who should have the least friction

in gaining access to premium content through using Gib, will have to install the extension on his browser before clicking on the link that Alice shares.

Given our findings, we see an opportunity to improve Bob's ease of use by replacing his extension with a web proxy. Overall, Charlie won't have to install a web component, Bob won't have to install an extension, and the gift card would be open-loop format.

IV. CONCLUSION

To summarize, we set out to research the implementation and associated challenges of an Internet gift card built on top of the Web Monetization API specification. To do so, we looked at using the Interledger Protocol, browser extensions, the Payment Handler specification, and the open-loop gift card model. We wanted to focus on ease of use for those receiving shared content and therein lay most of our challenges. The recipient of the gift card should be able to access the shared premium content as easily as clicking a URL, without going through any sign-up or download barriers. Additionally, we encountered uncharted territory researching the Web Monetization specification, as it is not yet a W3C Standard. Based on our findings, we suggest that one-time payments be added to the Web Monetization specification, to make it easier to handle payments from a user that wants to share premium content.

In conclusion, we have decided that a browser extension, with possible use of proxies, is the most feasible form factor of Gib in this early stage. Our next step will be to build out a proof-of-concept based on our analysis.

REFERENCES

- [1] [Web monetization](#). Adrian Hope-Bailie; Ben Sharafian; Nick Dudfield; W3C. 15 April 2021. W3C Working Draft. URL: <https://webmonetization.org/specification.html>.
- [2] [Payment Method Identifiers](#). Marcos Cáceres; Domenic Denicola; Zach Koch; Roy McElmurry. W3C. 05 September 2019. W3C Candidate. URL: <https://www.w3.org/TR/payment-method-id/#references>
- [3] [Payment Handler API](#). Adrian Hope-Bailie; Andre Lyver; Ian Jacobs; Rouslan Solomakhin; Jinho Bang. W3C. 21 October 2019. W3C Working Draft. URL: <https://www.w3.org/TR/payment-handler/>
- [4] [A Protocol for Interledger Payments](#). Stefan Thomas; Evan Schwartz. URL: <https://interledger.org/interledger.pdf>
- [5] [STREAM](#). URL: <https://interledger.org/rfcs/0029-stream>
- [6] "Gift Cards: Disclosure One Step Removed." Daniel R. Horne. *The Journal of Consumer Affairs*, vol. 41, no. 2, 2007, pp. 341–350. JSTOR, www.jstor.org/stable/23860062.
- [7] [Bring your payment method to the web with the Payment Handler API](#). Eiji Kitamura. Web Updates 2018. Google Developer Updates. URL: <https://developers.google.com/web/updates/2018/06/payment-handler-api>
- [8] [Web Component](#). URL: https://developer.mozilla.org/en-US/docs/Web/Web_Components

APPENDICES

Table A - Form factor metrics and considerations.

METRICS	Web Component Pros	Web Component Cons	Extension Pros	Extension Cons
Ease of Setup	Only the site manager must install			Everyone must install
Payment Availability		Micropayments initialized / web page	Micropayments initialized universally	
Browser Compatibility	Desktop & mobile browsers			Only desktop browsers
Anonymity		Difficult to keep user anonymous	Easy to keep user anonymous	