The Future of Payments
W3C Web Payments, and
The Interledger Protocol

Adrian Hope-Bailie
@ahopebailie
What is the future of payments?

Image credit: https://www.pexels.com/photo/smart-watch-smartwatch-futuristic-technology-9051/
How will we get there?
In the future there will be no payments...
Because in the future:

- Payment initiation will be **frictionless**
- Payment networks will be **interoperable**
- New **business models** will be viable
- New **ecosystems** will emerge
In the future there will be no payments... ...because payments will be completely UBIQUITOUS
Getting there...

1. Standardize the handshake
2. Standardize the messaging
3. Standardize the transport
The 4-corner model

Sender

Receiver
Payee centric model

Static

Dynamic

Sender

Receiver
Introducing Payment Apps

Payment Request API

Sender

Receiver

Photo credit: https://www.flickr.com/photos/78855484@N03/7223384344
Matchmaking based on payment methods

Sender

Payment App

Recipient

Website
The circle is complete
What if there is no common payment method?
We need a way to pay across networks?
And the answer is not another network

Sender
Payment App
Recipient
Website
It’s a network of networks
Lessons from the evolution of the Internet

RFC 1122 - Requirements for Internet Hosts -- Communication Layers
RFC 1123 - Requirements for Internet Hosts -- Application and Support
RFC 1009 - Requirements for Internet Gateways

Network Interface Layer
Internet Layer
Transport Layer
Application Layer

Ethernet
IP
HTTP
TCP
SMTP
UDP
FTP
RS232
X25
But, how do you actually **move** digital assets?
Well, digital assets are just balances in a ledger

<table>
<thead>
<tr>
<th></th>
<th>Sender</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receiver</td>
<td>0</td>
</tr>
</tbody>
</table>
So you just change the balances in the ledger

Sender | 50 | Recipient

Sender | 50
Receiver | 50
Ledger protocols move assets in a ledger
But not everyone has accounts on the same ledger.
We need a connector that has accounts on both ledgers. A digital asset “switch”.
The connector accepts a transfer on one ledger in exchange for making a transfer on another.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sender</strong></td>
<td>100</td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>100</td>
</tr>
<tr>
<td><strong>Receiver</strong></td>
<td>0</td>
</tr>
</tbody>
</table>
The result is that the assets **move** from the sender’s account to the receiver’s.
The Interledger protocols define how connectors route and transfer digital assets between ledgers

Interledger Layer

Ledger Interface Layer

ILP

SLP

Bitcoin

...
But how can we be sure the connector won't drop the transfer?
Different transport layer protocols offer different delivery guarantees.
Optimistic Transfer Protocol is... optimistic

- High volume, extremely low value use cases
- Micropayments
To improve on OTP we need **atomicity**.
Either all transactions complete...
...Or none of them do

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sender</strong></td>
<td>100</td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>100</td>
</tr>
<tr>
<td><strong>Receiver</strong></td>
<td>0</td>
</tr>
</tbody>
</table>
This problem is commonly solved using the two phase commit pattern for transaction atomicity.
To support the Universal Transfer Protocol, ledgers must be capable of staging a transfer through escrow.
The payment is prepared by putting assets in **escrow** on each ledger and all ledgers agreeing on a **release condition** proposed by the receiver.
The payment is executed by **releasing funds** to the **receiver first** and then passing the signed release fulfillment back down the line.

<table>
<thead>
<tr>
<th></th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sender</strong></td>
<td>75</td>
</tr>
<tr>
<td><strong>Escrow</strong></td>
<td>25</td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>75</td>
</tr>
<tr>
<td><strong>Escrow</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Receiver</strong></td>
<td>25</td>
</tr>
</tbody>
</table>
Connectors have an incentive to pass the fulfillment proof to the next and get paid.
The Atomic Transfer Protocol uses trusted **notaries** to trigger the execution of the transfers.
These basic building blocks enable digital assets to be securely relayed across multiple ledgers and networks.
Chained Payments
From any sender to any receiver through one or more connectors
Limitlessly Scalable

Connectors and Ledgers can be added to handle more payments
Connecting Disparate Systems

Minimal standard to link banks, networks, telcos, etc

Central Bank

Payment Network

Mobile Network Operator
Free the world’s liquidity
Unlocking liquidity from multiple sources to lower capital costs
Use case specific application layer protocols can be developed on top of the Interledger stack.
Open Web Payment Scheme provides basic consumer payments across networks
Payment Apps + Interledger = ?
Payment Apps + Interledger = ?

- Autonomous payments and the Internet of Things
  - Massive increase in global payments volume
  - Financial incentives driving new behaviour

- Micropayments-based business models
  - Goodbye advertising and DRM
  - Privacy and convenience... together

- Personal ledgers
Increasing speed increases the volume

Information exchange exploded because of the Internet’s speed and reach
402
Payment Required
This code is reserved for future use

https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.3
402
Payment Required
This code is ready for use

https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.3
Thank You

Adrian Hope-Bailie
@ahopebailie

Web Payments Activity
https://w3.org/payments

Interledger Protocol
https://interledger.org