**Lightning and Interledger - SWOT**

**Goals**

* **Lightning** - Scale Bitcoin’s transaction throughput and lower transaction latency
* **Interledger** - Connect all currencies and ledgers in the world
* **Both** - Enable fast, cheap payments and micropayments

**Technical Starting Point**

* **Both** - Chaining hashlocked transfers together enables multi-hop payments through untrusted intermediaries

**Privacy vs Efficiency**

* **Lightning** - All transactions are onion-routed by default to keep user transaction history private even from the nodes routing the payments. As a result, packets must be source routed, meaning every node needs a full and up-to-date view of the entire network
* **Interledger** - ILP packets are meant to take the most efficient path through the network and are not relayed through more hops than needed. Like the Internet, routing decisions are made at each hop so nodes do not need a full view of the network. Onion routing or mix networks can be built on top of the core protocol, just as TOR is built on top of the Internet.

**Currency / Asset Support**

* **Lightning** - Arguably, Lightning has many design choices that make it more suited to a single-asset network than a multi-asset network. For example, the fact that all nodes need an up-to-date view of the network may be feasible with a single currency and infrequently updated prices, but it is infeasible for constantly changing cross-currency exchange rates. Additionally, Lightning’s long hashlock timeouts would create a strong incentive to exploit the “free option” problem and steal from nodes providing cross-currency liquidity. For more details see [HTLCs Considered Harmful 2](https://forum.interledger.org/t/htlcs-considered-harmful/81)and this thread from the Lightning mailing list on [An Argument for Single-Asset Lightning Network 1](https://lists.linuxfoundation.org/pipermail/lightning-dev/2018-December/001752.html).
* **Interledger** - Many of the design choices that shaped Interledger were specifically intended to enable it to efficiently and safely handle multiple assets. For example, Interledger uses Internet-style non-source routing so that nodes do not need to constantly broadcast their changing exchange rates throughout the network. Also, Interledger switched from ledger-enforced hashlocks to connector-enforced hashlocks to reduce the required timeouts and minimize the free option problem. See [Simplifying Interledger: The Graveyard of Possible Protocol Features 2](https://medium.com/interledger-blog/simplifying-interledger-the-graveyard-of-possible-protocol-features-b35bf67439be) for more details on the tradeoffs and logic behind them.

**Ledger Integrations**

* **Lightning** - The protocol is [defined in terms of Bitcoin scripts 3](https://github.com/lightningnetwork/lightning-rfc/blob/6508af51459a6c128ce2672df62480abf9b35c07/03-transactions.md) and specific features have been required and proposed for the Bitcoin ledger in order to support Lightning, such as [SegWit](https://www.coindesk.com/information/what-is-segwit) and [SIGHASH\_NOINPUT 1](https://www.coindesk.com/new-twist-lightning-tech-coming-soon-bitcoin). Connecting Lightning to ledgers that are not forks of the Bitcoin codebase would likely require reimplementing Bitcoin scripts on top of the other ledger.
* **Interledger** - Designed from the start to connect all of the blockchains and other types of ledgers in the world, Interledger was meant to be completely agnostic to specific ledger features and instead was built upon a ledger abstraction. As it developed, the abstraction was simplified to the point where the only feature required to integrate a ledger is that it supports simple transfers between accounts (and even cash – a “ledger” without an API – can be used).

**Collateralization and Credit**

* **Lightning** - All payment channels are fully collateralized and each node must tie up liquidity in each channel to cover the total value of transactions that can be routed through that channel without balanced bilateral flows or an on-chain transaction.
* **Interledger** - Each bilateral link may be fully collateralized or may operate on limited credit lines. Importantly, the chained hashlocks used by both Interledger and Lightning isolate risks so that each node can only be affected by the arrangements they make with their directly connected peers. See [Hashed Timelock Agreements (HTLAs) 2](https://interledger.org/rfcs/0022-hashed-timelock-agreements/) for more details, but note that on-ledger escrow is no longer used due to performance reasons.

**Atomicity vs Packetization**

Each bilateral link or payment channel has a limited capacity based on how much liquidity (money) the parties allocate to that link. There are different strategies for enabling larger payments to be sent through links with lower capacity:

* **Lightning** - With the proposed [Atomic Multi-Path (AMP) Payments](https://lists.linuxfoundation.org/pipermail/lightning-dev/2018-February/000993.html) approach, a larger payment sent through Lightning would be split into multiple concurrent transfers and each “split” would be held by the receiver until they receive the total amount.
* **Interledger** - Like the Internet, Interledger splits larger transfers into smaller packets and sends them as quickly as the network can support (meaning it could be all at once or one at a time). The parallel is so strong that the [STREAM 2](https://medium.com/interledger-blog/streaming-money-and-data-over-ilp-fabd76fc991e) protocol built on ILP uses congestion control algorithms directly inspired by those used in the Internet’s TCP. Interledger prioritizes fast, simple forwarding to avoid allowing nodes to tie up large amounts of the network’s liquidity for payments that may or may not be executed.

**Layer 2 vs Layer 3**

* **Lightning** - The network is most often described as a “Layer 2” technology built on top of Bitcoin (the “Layer 1” blockchain).
* **Interledger** - The protocol can be used to connect different Layer 2 scaling technologies, including but not limited to the Lightning Network, [Ethereum’s Raiden Network 2](https://raiden.network/), and [XRP Payment Channels 4](https://developers.ripple.com/payment-channels.html). The core Interledger Protocol (ILP) is a “Layer 3” protocol, and the Interledger stack includes Layer 4 and 5 protocols built upon it. See [Layer 3 Is for Interoperability 7](https://medium.com/xpring/layer-3-is-for-interoperability-ca387fa5f7e2) for more details of the stack.