



Generalizing BlockSci to Cross-Chain Analyses of Forked Ledgers

Master thesis

Inday Students, December 2nd 2020

Martin Plattner, Universität Innsbruck
martin@mplattner.at



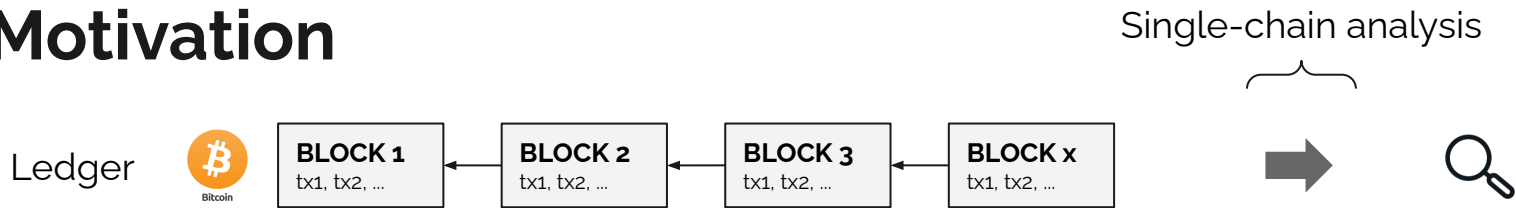
Motivation



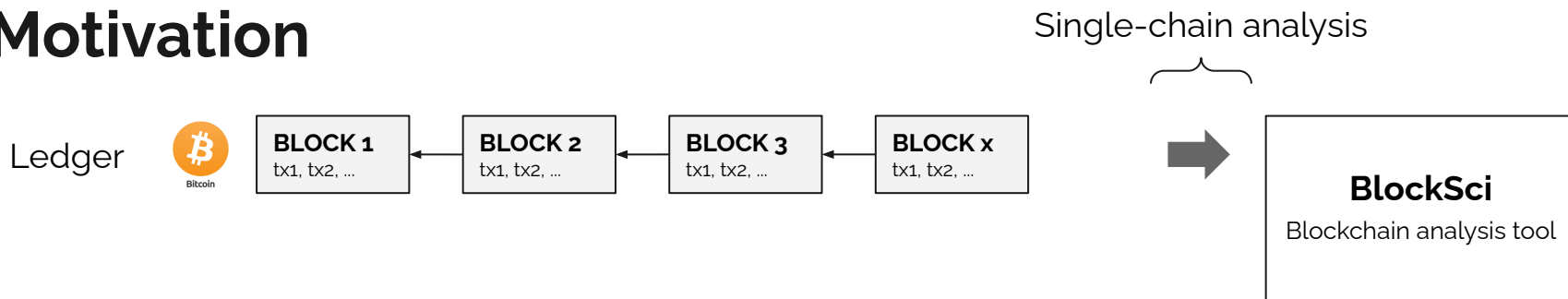
Motivation



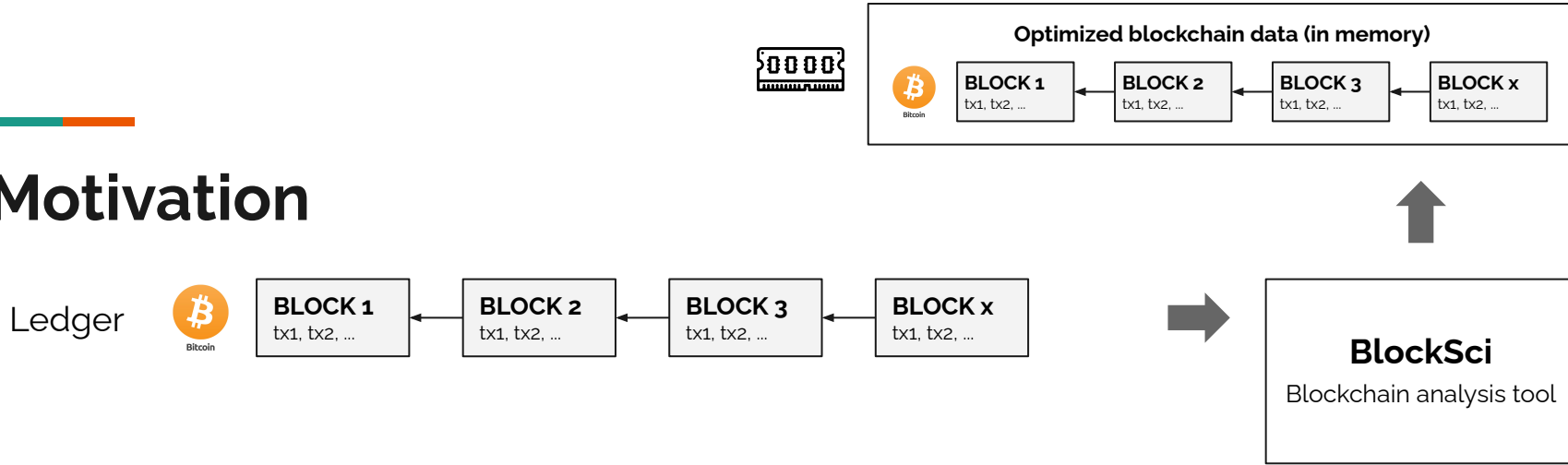
Motivation



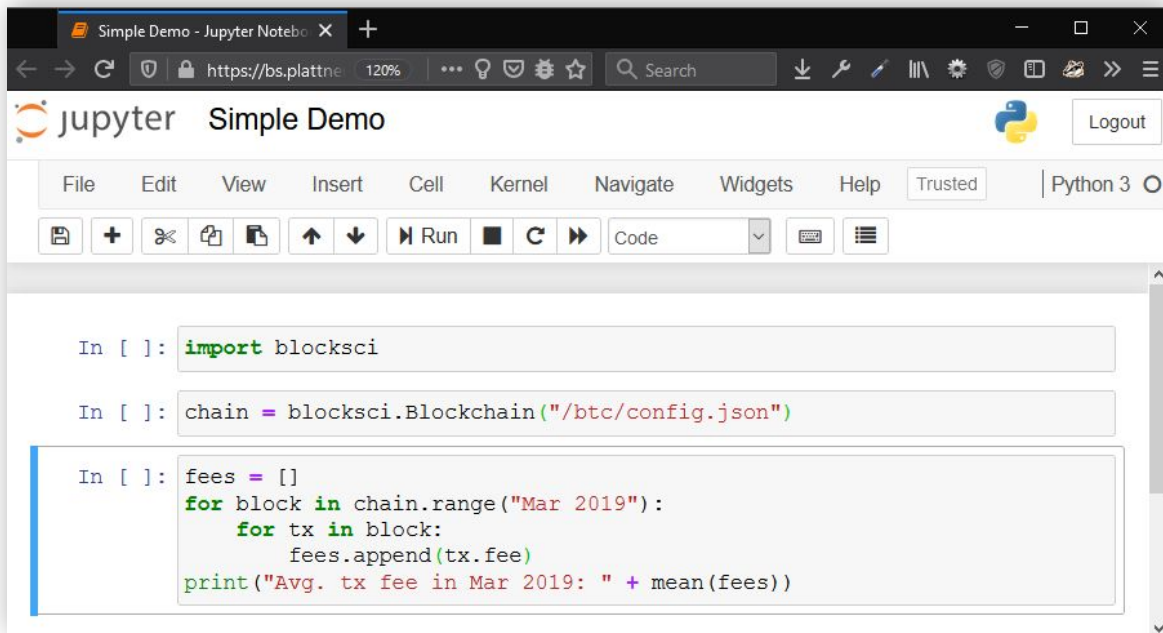
Motivation



Motivation



Motivation



```
In [ ]: import blocksci

In [ ]: chain = blocksci.Blockchain("/btc/config.json")

In [ ]: fees = []
        for block in chain.range("Mar 2019"):
            for tx in block:
                fees.append(tx.fee)
        print("Avg. tx fee in Mar 2019: " + mean(fees))
```



Optimized blockchain data (in memory)



BLOCK 1

tx1, tx2, ...

BLOCK 2

tx1, tx2, ...

BLOCK 3

tx1, tx2, ...

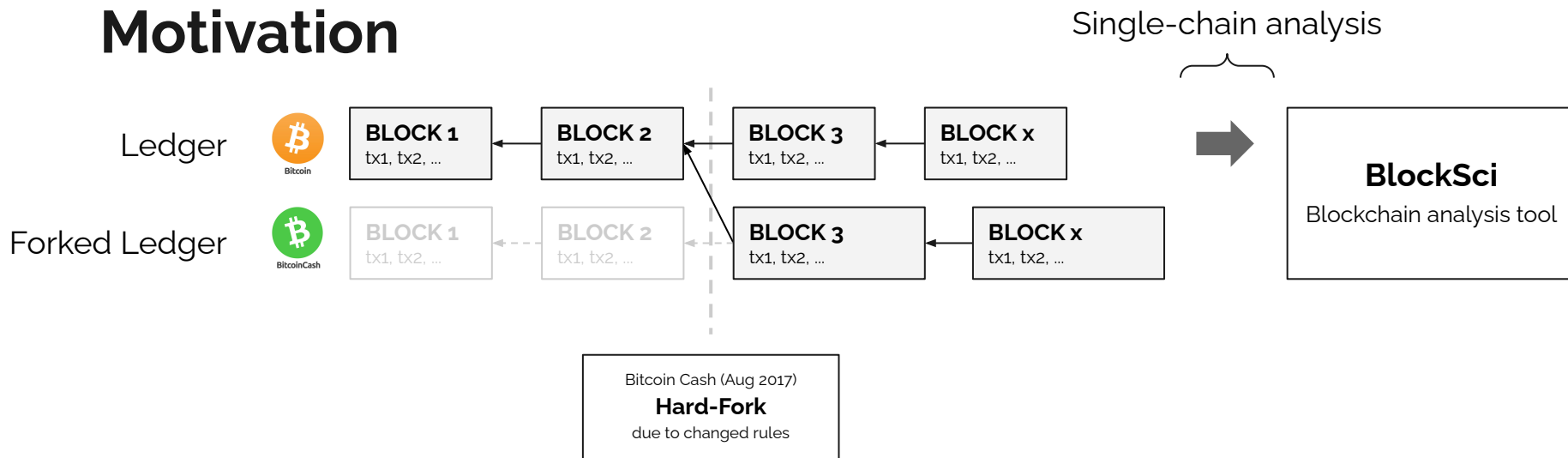
BLOCK x

tx1, tx2, ...

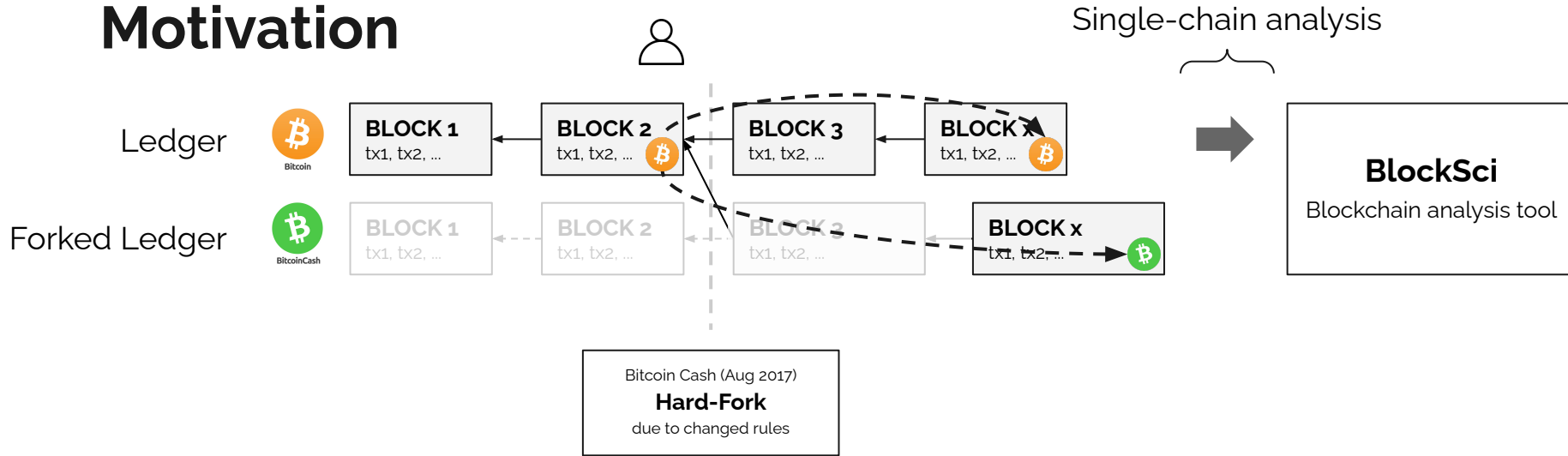
BlockSci

Blockchain analysis tool

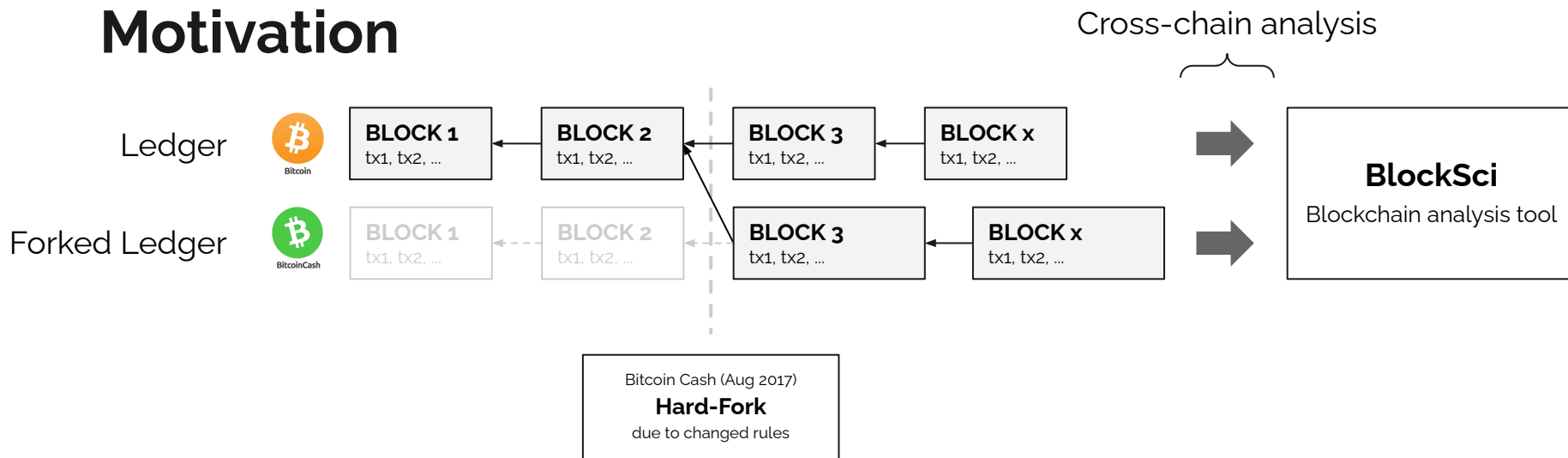
Motivation



Motivation

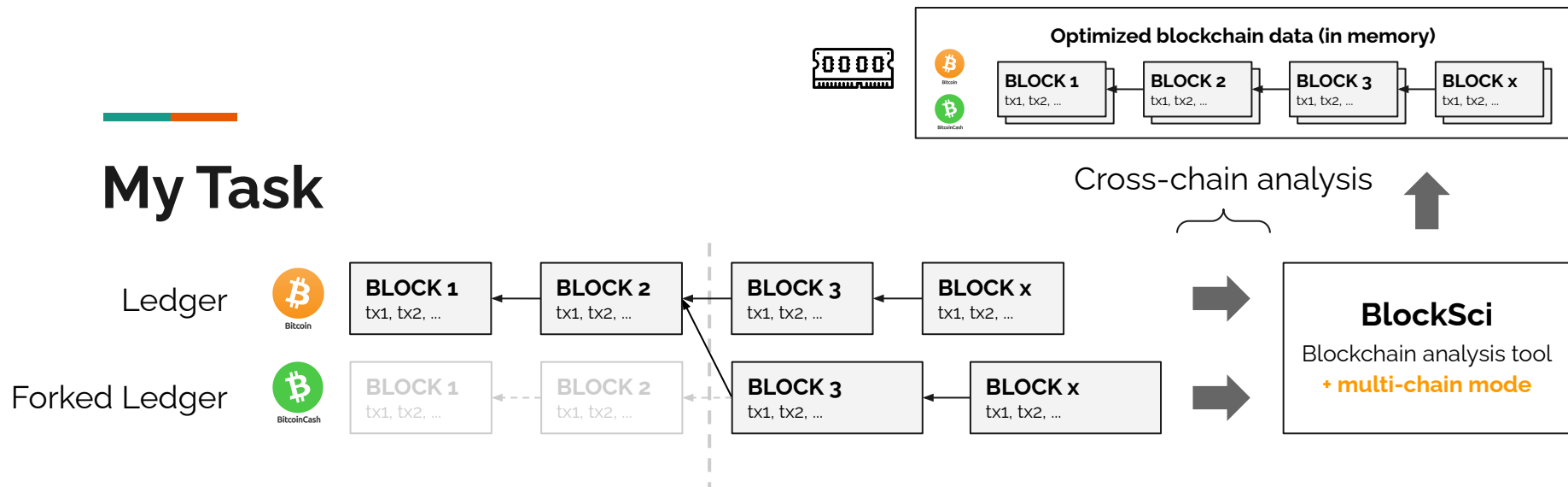


Motivation



Generalizing BlockSci to Cross-Chain Analyses of Forked Ledgers

My Task



Main contribution

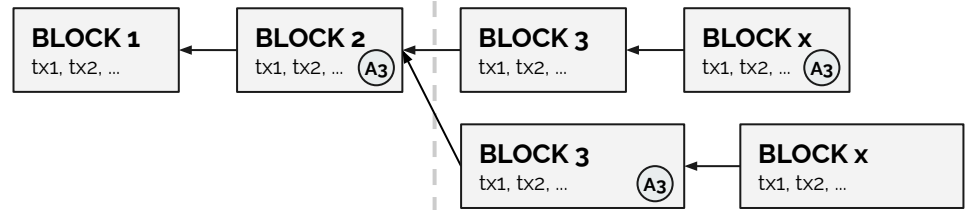
Multi-chain mode for BlockSci



Requirements

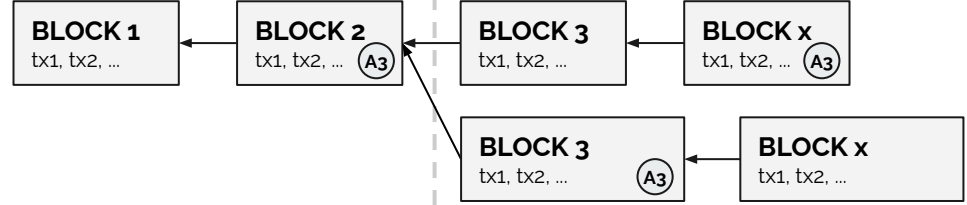
Type	No.	Requirement	Priority
Functional	1	Normalized addresses	MUST
	2	BTC and BCH support	MUST
	3	Flexible configuration	MUST
	4	Anticipate cross-chain queries	MUST
Non-functional	1	Optimize memory consumption	MUST
	2	Maintain high performance	SHOULD
	3	Backwards compatibility (API)	SHOULD
	4	Extensibility	SHOULD

Requirements



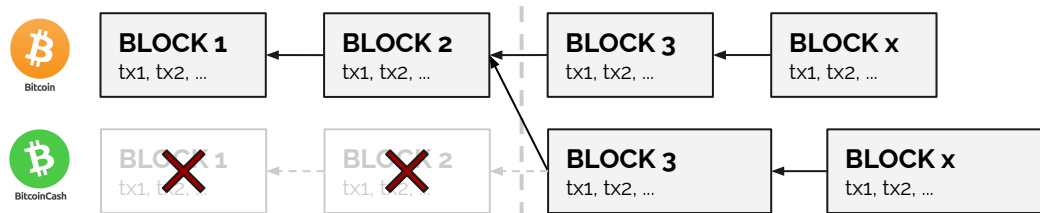
Type	No.	Requirement	Priority
Functional	1	Normalized addresses	MUST
	2	BTC and BCH support	MUST
	3	Flexible configuration	MUST
	4	Anticipate cross-chain queries	MUST
Non-functional	1	Optimize memory consumption	MUST
	2	Maintain high performance	SHOULD
	3	Backwards compatibility (API)	SHOULD
	4	Extensibility	SHOULD

Requirements



Type	No.	Requirement	Priority
Functional	1	Normalized addresses	MUST
	2	BTC and BCH support	MUST
	3	Flexible configuration	MUST
	4	Anticipate cross-chain queries	MUST
Non-functional	1	Optimize memory consumption	MUST
	2	Maintain high performance	SHOULD
	3	Backwards compatibility (API)	SHOULD
	4	Extensibility	SHOULD

Requirements



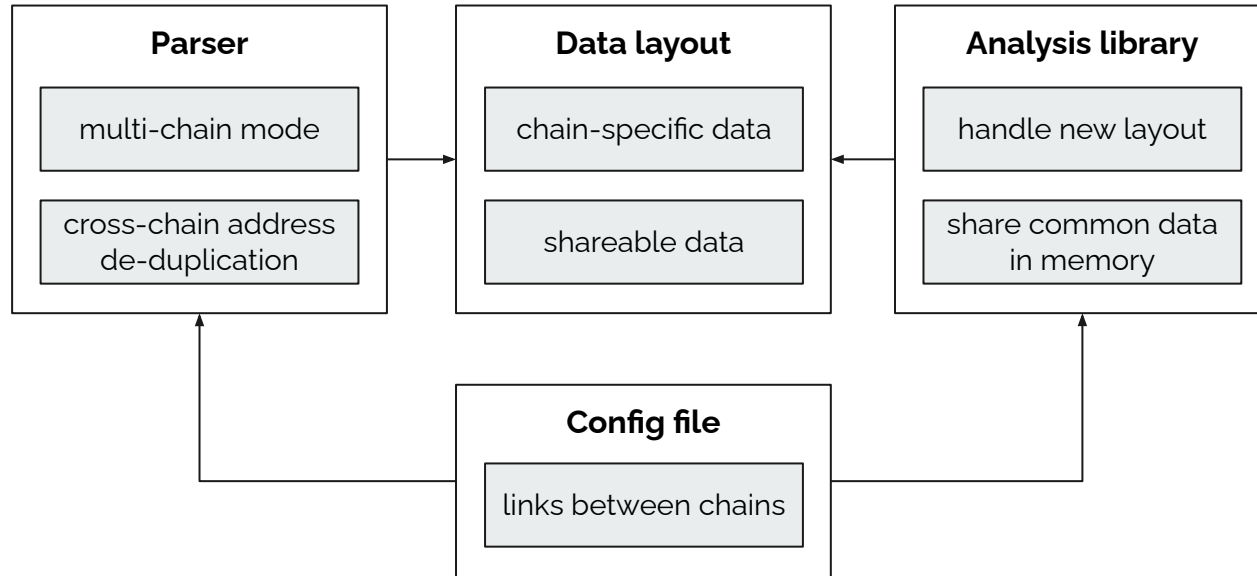
Type	No.	Requirement	Priority
Functional	1	Normalized addresses	MUST
	2	BTC and BCH support	MUST
	3	Flexible configuration	MUST
	4	Anticipate cross-chain queries	MUST
Non-functional	1	Optimize memory consumption	MUST
	2	Maintain high performance	SHOULD
	3	Backwards compatibility (API)	SHOULD
	4	Extensibility	SHOULD



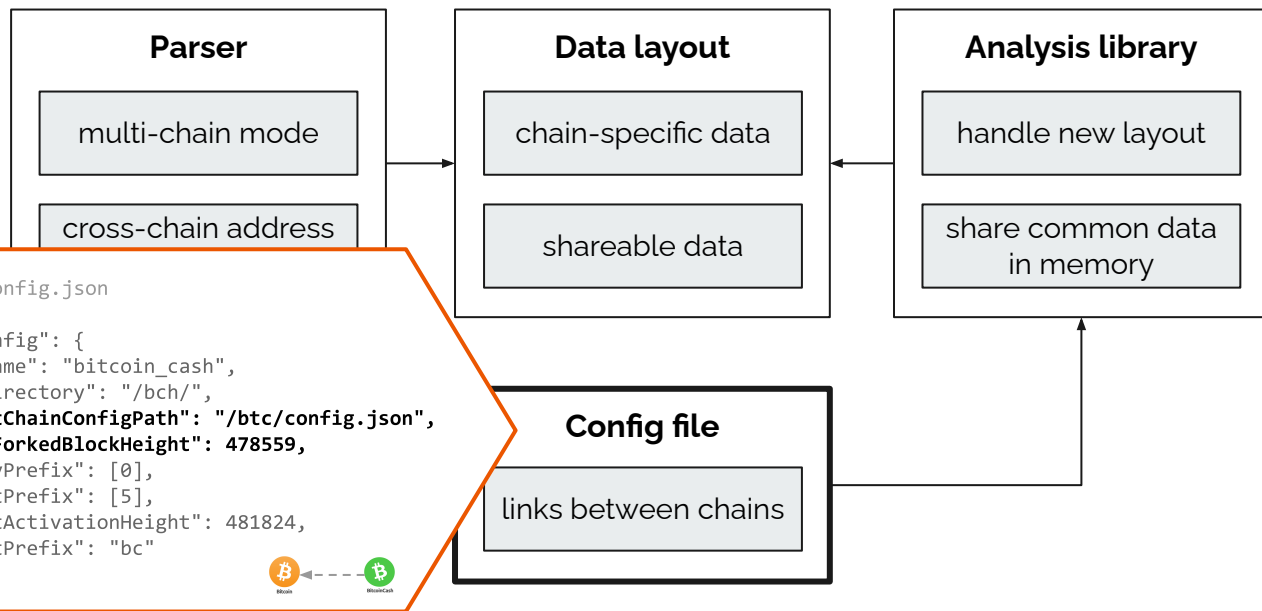
Requirements

Type	No.	Requirement	Priority
Functional	1	Normalized addresses	MUST
	2	BTC and BCH support	MUST
	3	Flexible configuration	MUST
	4	Anticipate cross-chain queries	MUST
Non-functional	1	Optimize memory consumption	MUST
	2	Maintain high performance	SHOULD
	3	Backwards compatibility (API)	SHOULD
	4	Extensibility	SHOULD

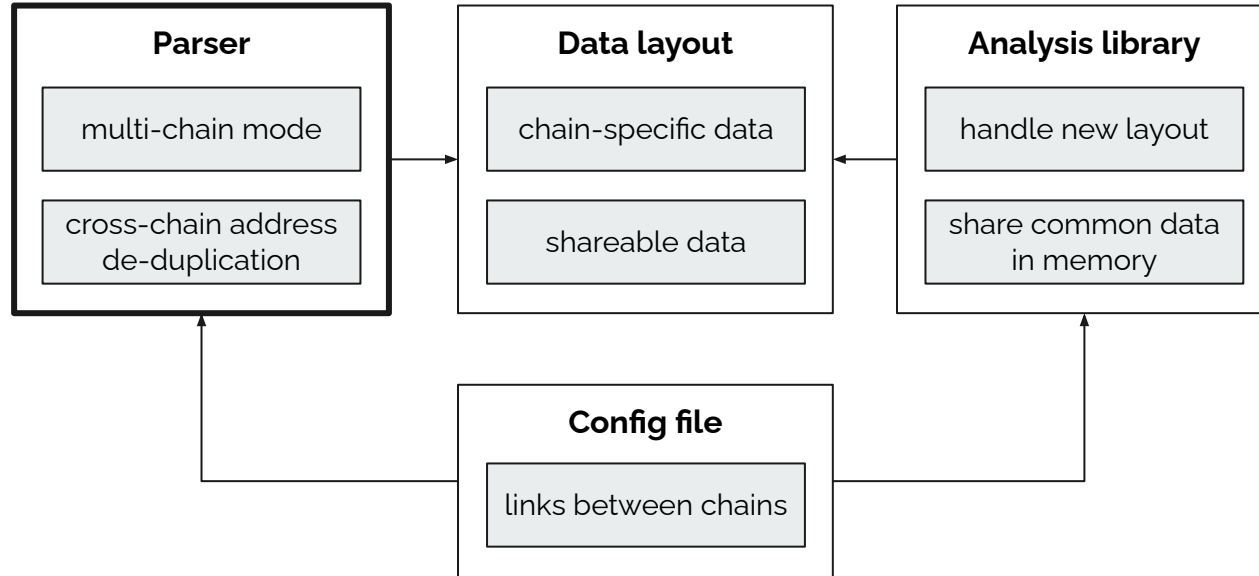
Required Changes



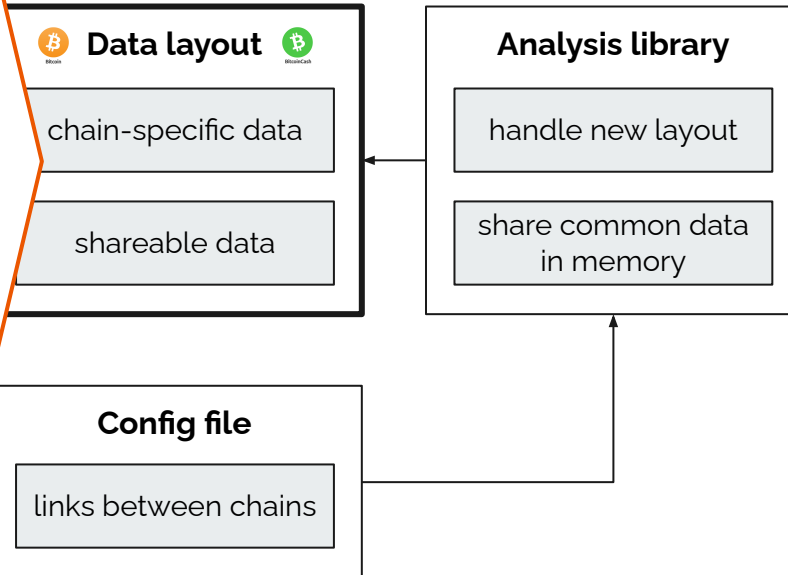
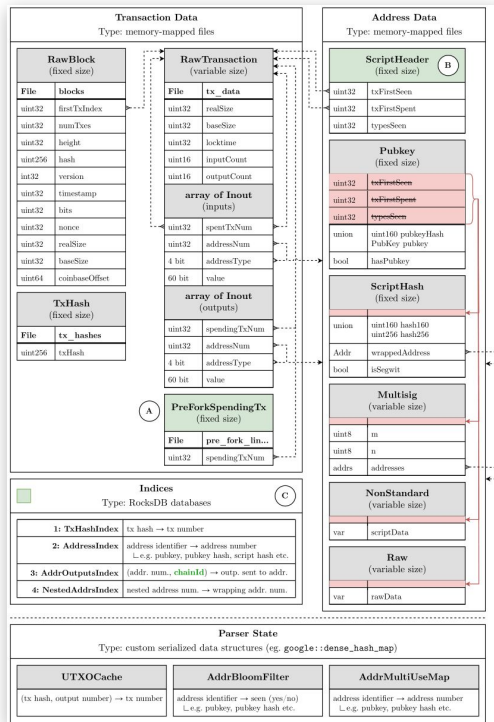
Required Changes



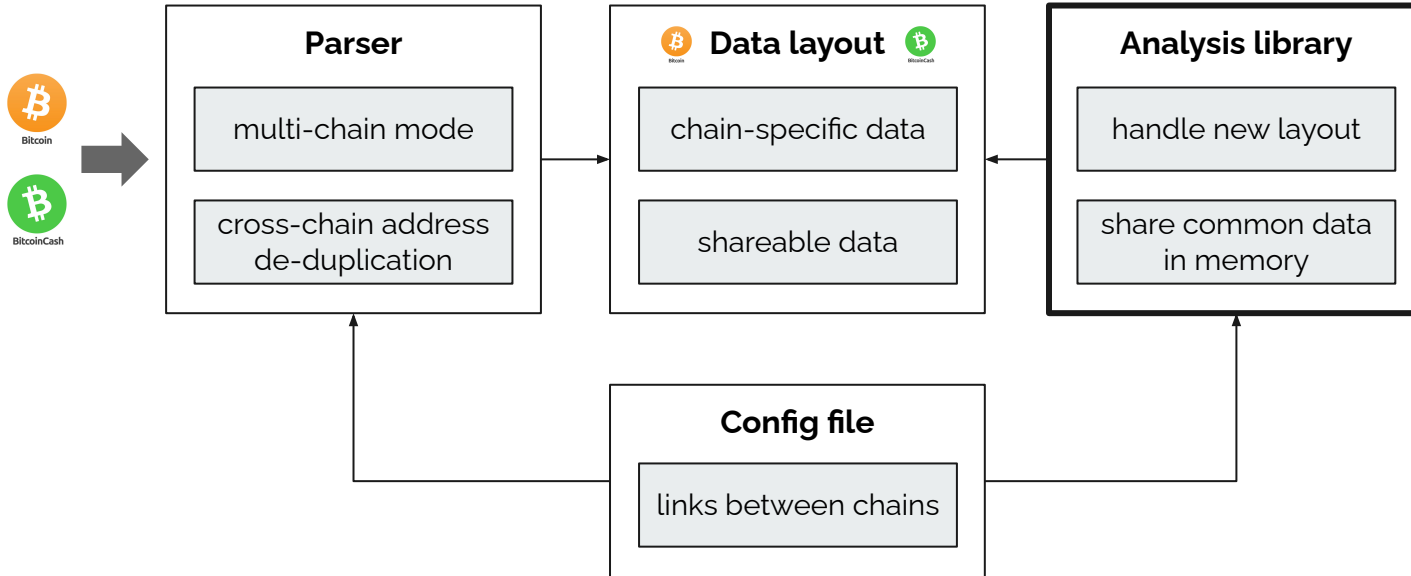
Required Changes



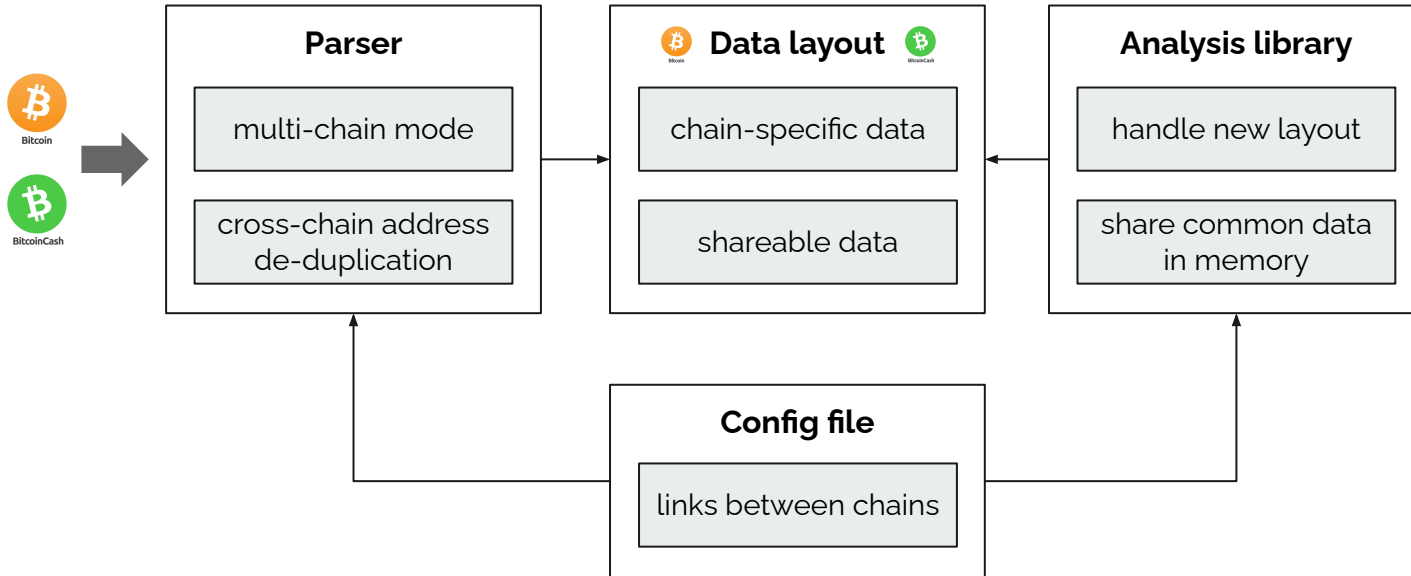
anges



Required Changes



Required Changes

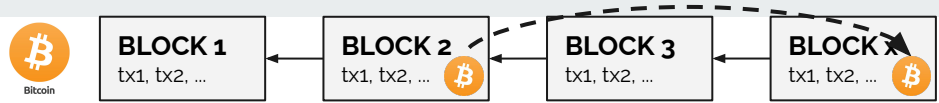




Cross-Chain Address Clustering

BTC
single-chain clustering

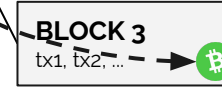




Cross-Chain Address Clustering

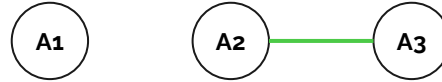
BTC
single-chain clustering





Cross-Chain Address Clustering

BCH
single-chain clustering



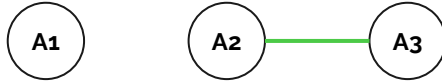
BTC
single-chain clustering





Cross-Chain Address Clustering

BCH
single-chain clustering



BTC
single-chain clustering



Addresses are normalized in the new multi-chain mode

Cross-Chain Address Clustering



BLOCK 1

tx1, tx2, ...

BLOCK 2

tx1, tx2, ...

BLOCK 3

tx1, tx2, ...

BLOCK x

tx1, tx2, ...



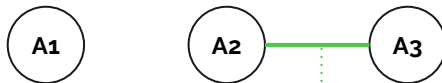
BLOCK 3

tx1, tx2, ...

BLOCK x

tx1, tx2, ...

BCH
single-chain clustering



BTC
single-chain clustering

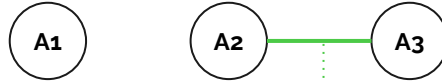


BTC ← BCH
cross-chain clustering



Cross-Chain Address Clustering: Results

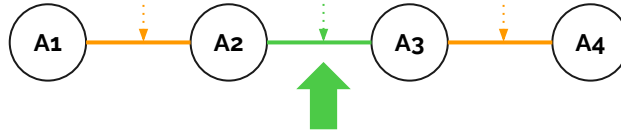
BCH
single-chain clustering



BTC
single-chain clustering

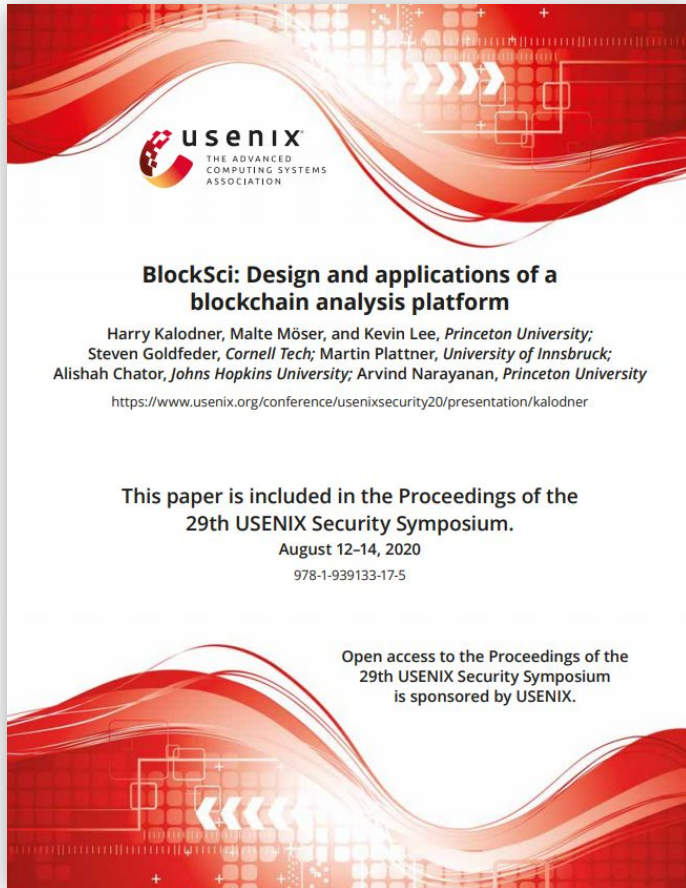


BTC ← BCH
cross-chain clustering



1.05 million additional cluster merges (new edges)

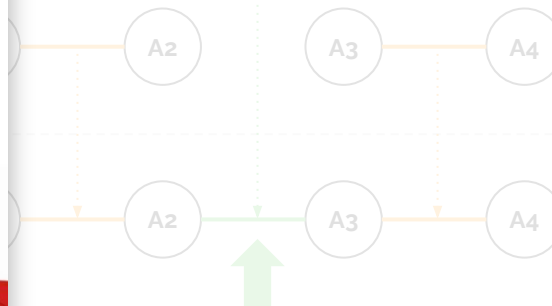
Affects 100,000 users with 30 million addresses (3% of all Bitcoin addresses)



Kalodner, H., Möser, M., Lee, K., Goldfeder, S., Plattner, M., Chator, A., & Narayanan, A. (2020).

Blocksci: Design and applications of a blockchain analysis platform.

In 29th USENIX Security Symposium (pp. 2721-2738).



Final cluster merges (new edges) as of Dec 2017

Nodes with 30 million addresses (3% of all Bitcoin addresses)



Conclusion

- Forks offer a powerful data source due to the links between chains
- Implemented multi-chain mode for cross-chain analyses of forked ledgers
- Implemented cross-chain address clustering
 - Found that privacy of users can be hurt across chains

Thanks.

Questions, thoughts, feedback?

martin@mplattner.at

More details, full thesis, code, and paper:

<https://mplattner.at>

Acknowledgments: public funding for my work

1. TITANIUM: Tools for the Investigation of Transactions in Underground Markets. (EU Horizon 2020)
2. VIRTCRIME: Forensic Methods and Solutions for the Analysis of Criminal Transactions in Post-Bitcoin Cryptocurrencies. (Austrian Research Promotion Agency FFG)