Bitcoin Evolution and Challenges

Blockchain Evolution

1. Explain the implications of changing consensus-relevant methods or data structures. Decide if the following changes to the Bitcoin software would impact the consensus-layer.

   - Transactions in the mempool are deleted after a certain elapsed time.
   - The scheme for transactions is changed such that the transaction fee is explicitly stated.
   - After receiving and validating a block, the node encrypts the data before storing locally off-chain. (The data is decrypted before being sent to other nodes)
   - The node enables a new method / RPC-call, in which the user can search for stored texts on the Blockchain.
   - Bitcoin Script now supports an Op-Code which introduces loops and jumps.
   - The block size is increased from 1 MB to 1.5 MB.

Generally, changes to the consensus layer implicate the necessity for a hard- or a soft-fork. The complete network needs to upgrade to the new version, otherwise changes could lead to two seperate chains.

- Transactions in the mempool are deleted after a certain elapsed time. No, this change does not impact the consensus layer. If a single node removes transactions after a certain period of time, it does not affect other nodes in their behavior. It is up to the node how it manages the mempool.
- The scheme for transactions is changed such that the transaction fee is explicitly stated. Yes. The transaction scheme is new and therefore other nodes have to recognize to format to validate transactions correctly.
- After receiving and validating a block, the node encrypts the data before storing. (The data is decrypted again before sent to other nodes). No. It is irrelevant for the network how a single node stores its data.
- The node enables a new method / RPC-call, in which the user can search for stored texts on the Blockchain. No. Advanced functionalities, which only read from one node are not impacting the consensus layer.
- Bitcoin Script now supports an Op-Code which introduces loops and jumps. Yes, new Op-Codes can only be used if the complete network agrees to the proposal. A node has to recognize and understand what an Op-Code does in order to execute it.
- The block size is increased from 1 MB to 1.5 MB. Yes. A new max-size for blocks requires an update for all nodes.

2. Assume that the Bitcoin development team plans to increase the maximum block size limit from 1MB to 10MB. Explain if this change requires a hard fork or soft fork and explain the risks of changing this property only.

This change can be considered a hard fork, as old version (legacy) becomes incompatible with the new one. Blocks produced by the new version nodes are not considered valid by the legacy nodes.
There are some risks:

- A hard fork can lead to two chains if the hash power of the new version miners is greater than the hash power of the legacy version miners. However, this can be dangerous when the legacy miners regain the majority hash power. In that case, it could be possible that the new chain (with 10MB blocks) gets wiped out as the miners of the new chain can also accept blocks of the legacy chain (miners follow the rule of "highest accumulated weight"). This would result in the 10MB blockchain being orphaned.
- There is a risk of a replay attack. If there are two chains, a user will control two types of coins, BTC_A and BTC_B. If the user creates a transaction spending his BTC_A, the transaction is also valid in BTC_B. Therefore, malicious nodes can propagate these transactions between the two networks, spending the coins from transactions on both chains.

Appropriate actions depend on the intentions of the development team: If they want to keep one blockchain and do not want to fork, then a compatible blockchain is the best shot. However, you must install replay protection if they are intentionally forking (like with Bitcoin Cash). The basic idea is that the chain compatibility is broken, such that BTC_A does not accept a block or transaction from BTC_B and the other way around.

3. In 2017, Bitcoin underwent the SegWit upgrade soft fork which enabled placing more transactions into a Bitcoin block without directly increasing the block size limit.

   (a) Briefly explain how SegWit manages to increase the transaction throughput without increasing the maximum block size.

SegWit introduced the concept of weight units (WU) and replaced the block size limit (1,000,000 bytes) with a block weight limit (4,000,000 WU). While non-SegWit transactions still take up the same space, SegWit transactions save up weight units as signature data is placed into the witness component, which weighs less than the rest of the transaction parts (*a witness data byte weighs a quarter of a non-witness data byte*).

$$Weight_{tx} = 4 * bytes_{nonwitness} + 1 * bytes_{witness}$$

$$\sum_{i=1}^{n} Weight_i \leq 4MWU$$

(b) What indicates that SegWit was a soft fork and not a hard fork?

SegWit was a soft fork as the new rules introduced by it (e.g., restricting miners to create blocks under 4,000,000 weight units) were not required to be followed by the legacy nodes (backward compatible). Thus, it was sufficient that only a majority of the network supported the changes (unlike a hard fork requiring the complete network to support the changes and upgrade the software to avoid a chain split).

SegWit avoided a hard fork by not increasing the block size (still 1MB), at least not in the perception of the legacy nodes. This is done by stripping away the witness data from SegWit transactions (while still keeping them valid by modifying *scriptSig* and *scriptPubKey*) before sending them to the legacy nodes. Read more about how legacy nodes verify SegWit transactions here: `https://medium.com/@BlockTalkChain/how-does-non-segwit-legacy-node-verify-segwit-transaction-c3bc0872842b`

If a block contains only non-SegWit transactions, it will have the same ($\leq$ 1MB) size for both legacy and SegWit nodes as a non-SegWit transaction weighs exactly 4x its size (hence, block weight = 4 x block size). However, if the block contains SegWit transactions, it will not have the same size for legacy and SegWit nodes. While SegWit nodes will perceive the block with the total raw size of the transactions, legacy nodes will see a version of the block with SegWit transactions stripped from their witness data (hence, have a smaller size).

With the increasing witness size, the difference between what a legacy node perceives as the size of a transaction and what a SegWit node computes will become more significant.

(c) Alice and Bob are two miners operating on the Bitcoin network. Bob believes that Bitcoin block size should be strictly limited to 1MB while Alice thinks this is a hard constraint for scaling the network. Thus, Bob ends up not adopting the SegWit upgrade, unlike Alice.
Let's assume that whenever a new block is mined, Alice and Bob compare the size of the block (in MB) they received from the peers nodes in the network. Name the two possible scenarios in terms of size comparison [$<$, $>$, $=$] and briefly explain how they can happen. Consider the transactions included in the block.

If all the transactions in the block are non-SegWit, the block will appear to have the same size for every miner. If there are SegWit transactions in the block, the block size will appear smaller for the non-SegWit miners.

Blockchain Attacks

1. Justify whether the following scenarios can be achieved by an attacker holding 51% of the network's hash power.

   - The attacker can block transactions from a single address.
   - The attacker can halt payments between some users.
   - The attacker can DoS the network.
   - The attacker can change the mining reward.
   - The attacker can create coins out of thin air.

   - The attacker can block transactions from a single address. Yes. If an honest node publishes a block which contains such transaction, the adversary can orphan this block.
   - The attacker can halt payments between some users. Yes. The attacker can prevent new transactions from gaining confirmations by modifying the ordering of transactions.
   - The attacker can DoS the network. Yes. The attacker can propose empty blocks and thus render the network unusable.
   - The attacker can change the mining reward. No. The attacker can't change the mining reward since this is a protocol level change.
   - The attacker can create coins out of thin air. No. The only way to mine/create tokens is by finding the next block.

2. Inform yourself about the 51% attack on Bitcoin Gold. Explain what happened and how high the damages were. Explain how exchanges can decrease the chance of such an attack.

   - May 18, the communications director of Bitcoin Gold alerts the crypto-community: Someone is trying to use 51% of hash power to perform double spends, advises to increase confirmations.
   - Suspected hacker has sent Bitcoin Gold to exchanges and trades them for other coins, withdraws the other coins. Double spends his transaction to the exchange and gains about 18 million USD.
   - To decrease the chances of a successful attack, exchanges have to, after receiving funds from unknown addresses, wait for a high number of confirmations, such it gets harder and harder for the attacker to double spend. The longer the transaction is in the blockchain, the harder it is to double spend it.

3. Selfish mining is a process in which an attacker with less than 50% of hashing power can attack the network. α defines the probability of the network choosing/following the block found by the attacker. Explain the minimum hash rate required to launch a successful attack if α is 100%.

If α is 100%, then whenever someone finds a block with the same height as the attacker, the attacker can publish his mined block and always win the race. If this is the case, then any amount of hashing power can execute selfish mining, as it has no downside. Since α is 100%, there is no chance of losing the race and, therefore, no risk.

To read more about selfish-mining, check this article written by Vitalik in 2013: `https://bitcoinmagazine.com/technical/selfish-mining-a-25-attack-against-the-bitcoin-network-1383578440`