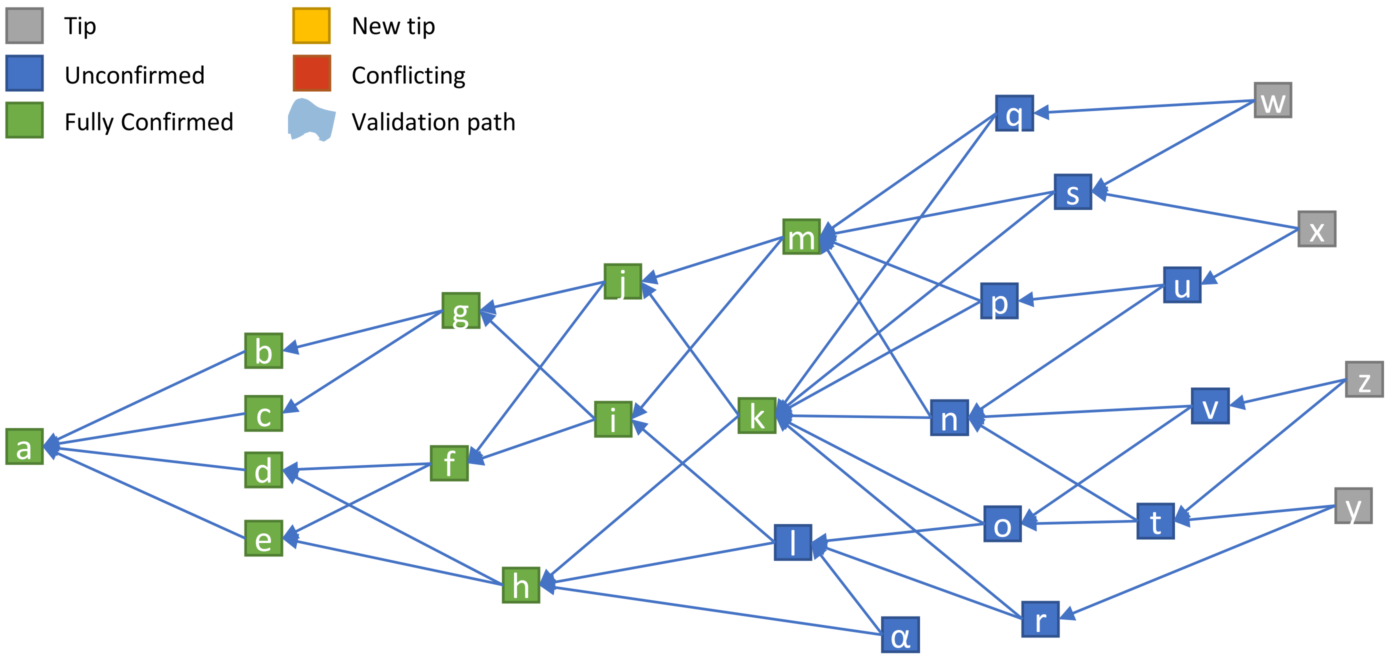
IOTA

Transactions, Confirmation And Consensus

IOTA Donations: QPLGOG9PMIMUAW9UDMUNZQHPXZPXDNGLBEIHILXHWHIOFHLIHPDDERXAJQKUQDEORMHSUWVZQE9JYSHIWADIIPAOJD

Initial Tangle State

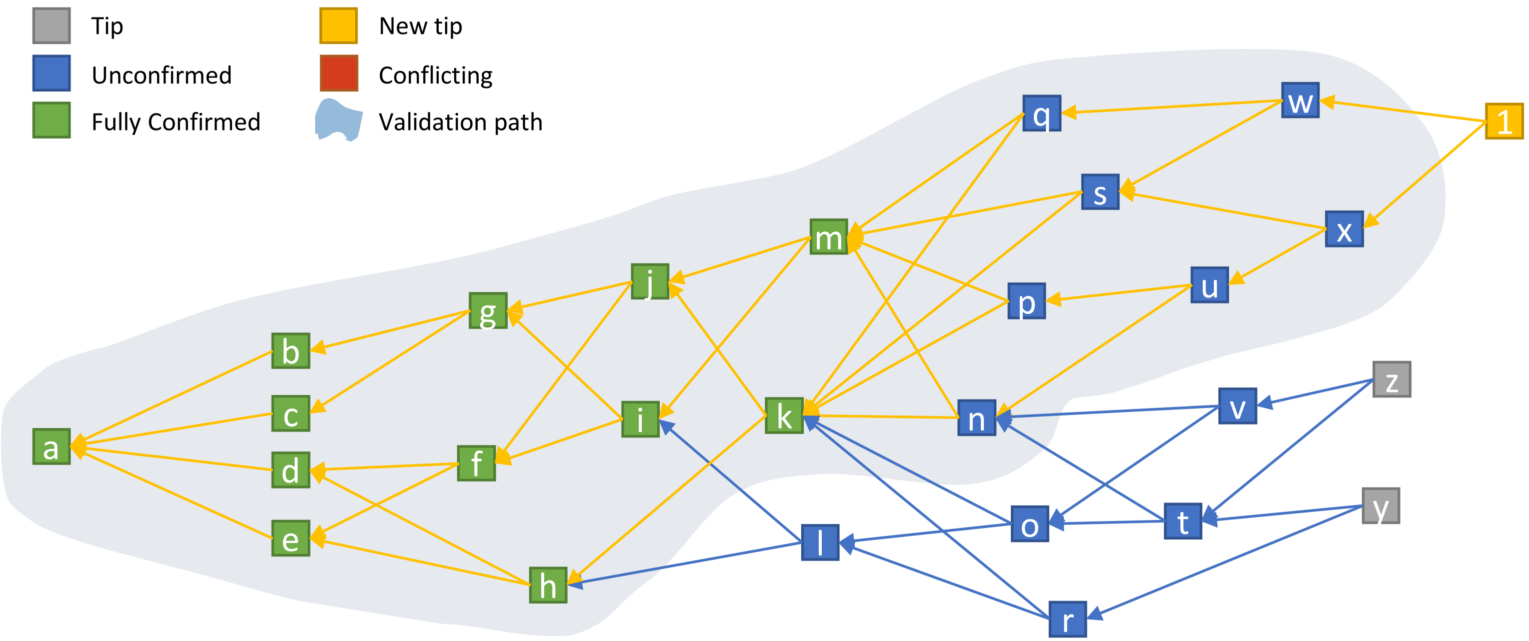


In contrast to blockchain technologies, IOTA does not build a clocked sequence of static blocks, each one containing a number of transactions. Instead, every single transaction can be attached to a by itself and in parallel to other transactions. The following slides will describe how adding transactions, validation and consensus works in IOTA.

The graph above shows a sample of a tangle which will be used in the subsequent slides to walk through some examples. In this and subsequent examples, green transactions are already confirmed by the network with high certainty (You'll find out why later. Spoiler: As with blockchain, it is about probabilities and there never going to be 100% carved-in-stone certainty), while the blue ones are only partially confirmed (with lower certainty). The grey (and later "yellow") boxes represent tips without any validation. Red transactions, on the later slides, are conflicting or invalid ones.

In the graph above transaction "α" is an example of an unusual transaction. It is referencing transaction „h" and „l". Since transaction „h" is already referenced by transaction „l", "α" would select one tip ("l") and one transaction that is obviously no tip at that time anymore ("h"). Such behavior seems to be no issue and tolerated by the network, currently.

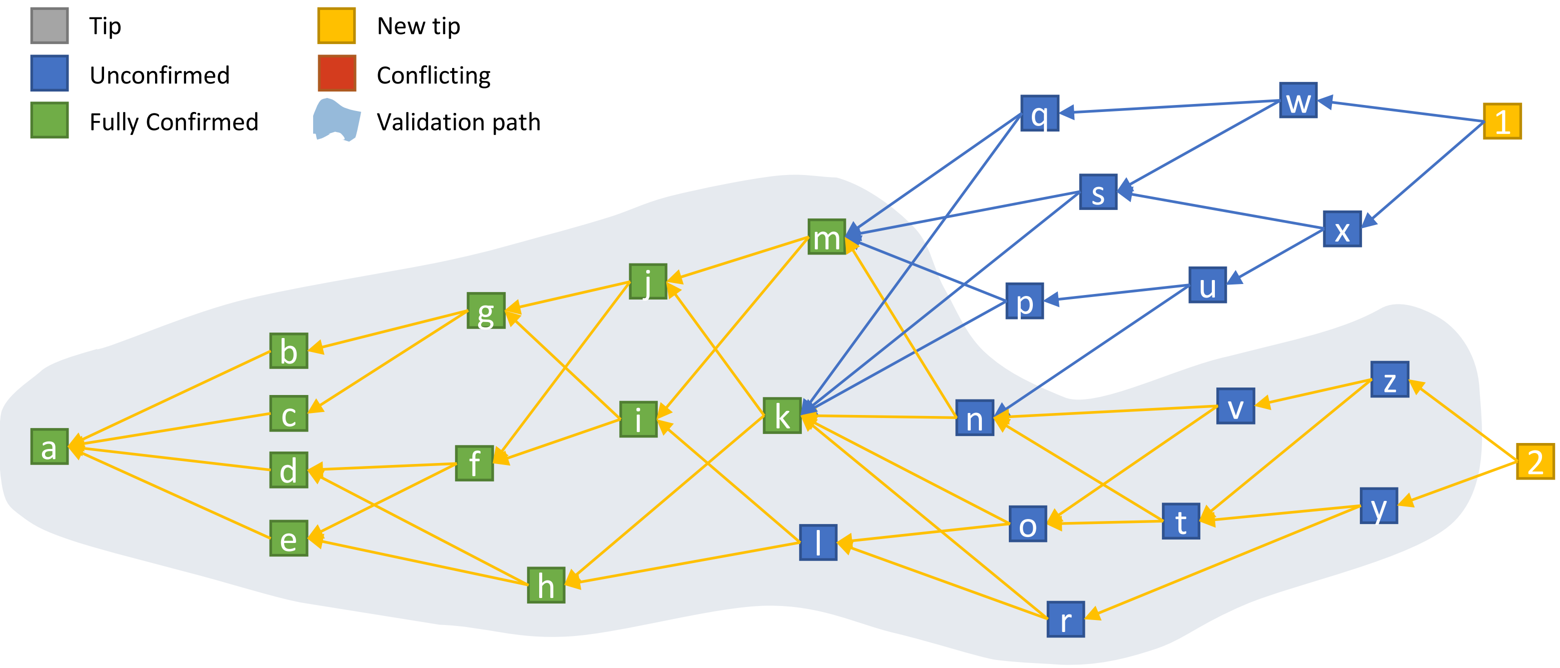
Adding A Transaction



In order to add a new transaction to the tangle, a user has to randomly pick two tangle tips (yet unconfirmed transactions) and validate them. Validating means that the user is checking the tip‘s signature, its PoW (little “Proof of Work” as spam protection) and makes sure that the tip is not in conflict with any of the previous (directly or indirectly referenced) transactions. If the chosen tips are legit, the user adds its new transaction by referencing them.

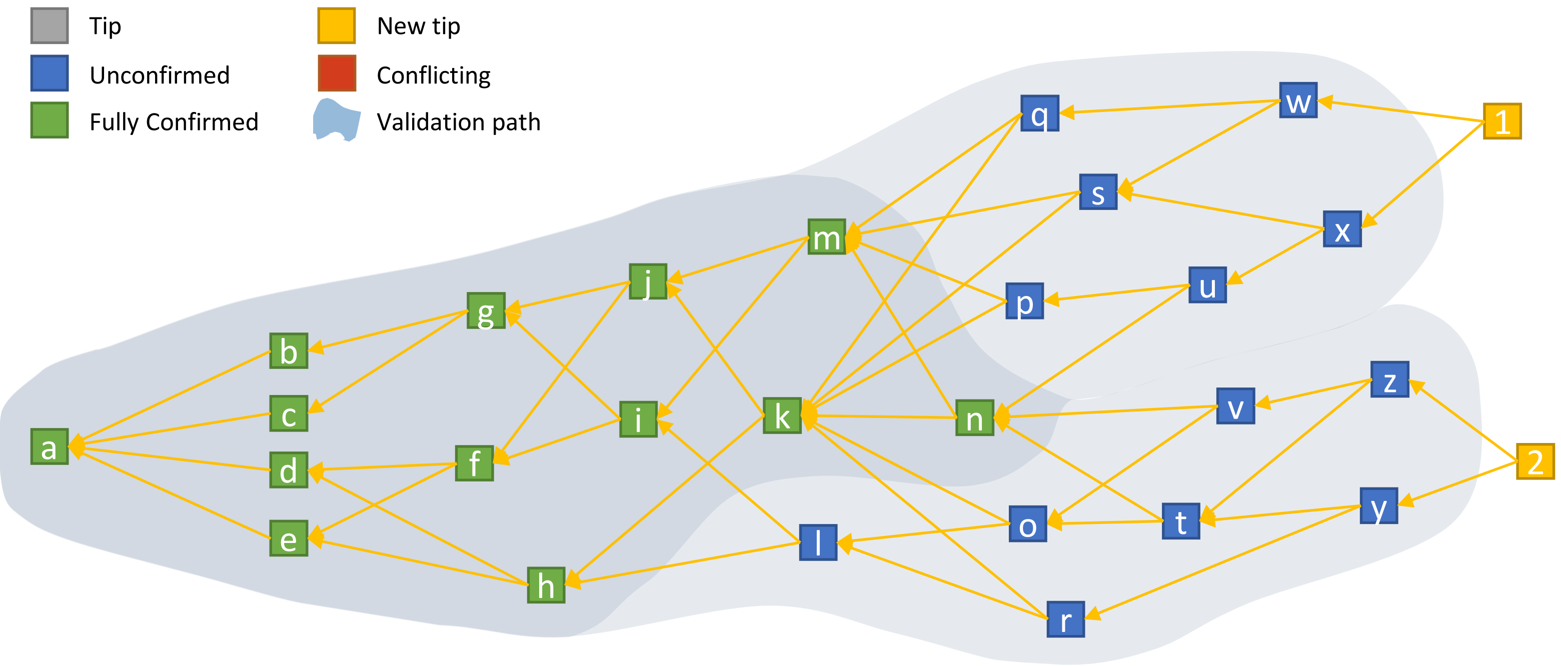
Transactions neither directly nor indirectly referenced by the chosen tips, are irrelevant for the current validation process. Somebody else, or a later transaction will take care of validating and knitting them into the tangle.

Another Transaction



At the same time (or earlier or later, whatever) another user might be about to add its new transaction in a different position. It chose the tips “z” and “y”. By doing so, it is validating a large portion of the same transactions as already validated via transaction „1“ (“a” to “k”, “m” and “n”), plus some additional ones that were not in the validation path of transaction „1“ (“l”, “o”, “r”, “t”, “v”, “y” and “z”).

New Tangle State



By overlaying the validation paths of transaction “1” and “2”, we can see that some of the transactions are only confirmed by one, while others were confirmed by both of them. Transactions validated and confirmed by all of the current tips are considered fully confirmed. Hence, transactions “n” moved deeper into the tangle and turned green now. All subsequent transactions attaching to “1” and/or “2” or its children or theirs again (and so on) will keep re-validating and confirming transaction “n” from now on.

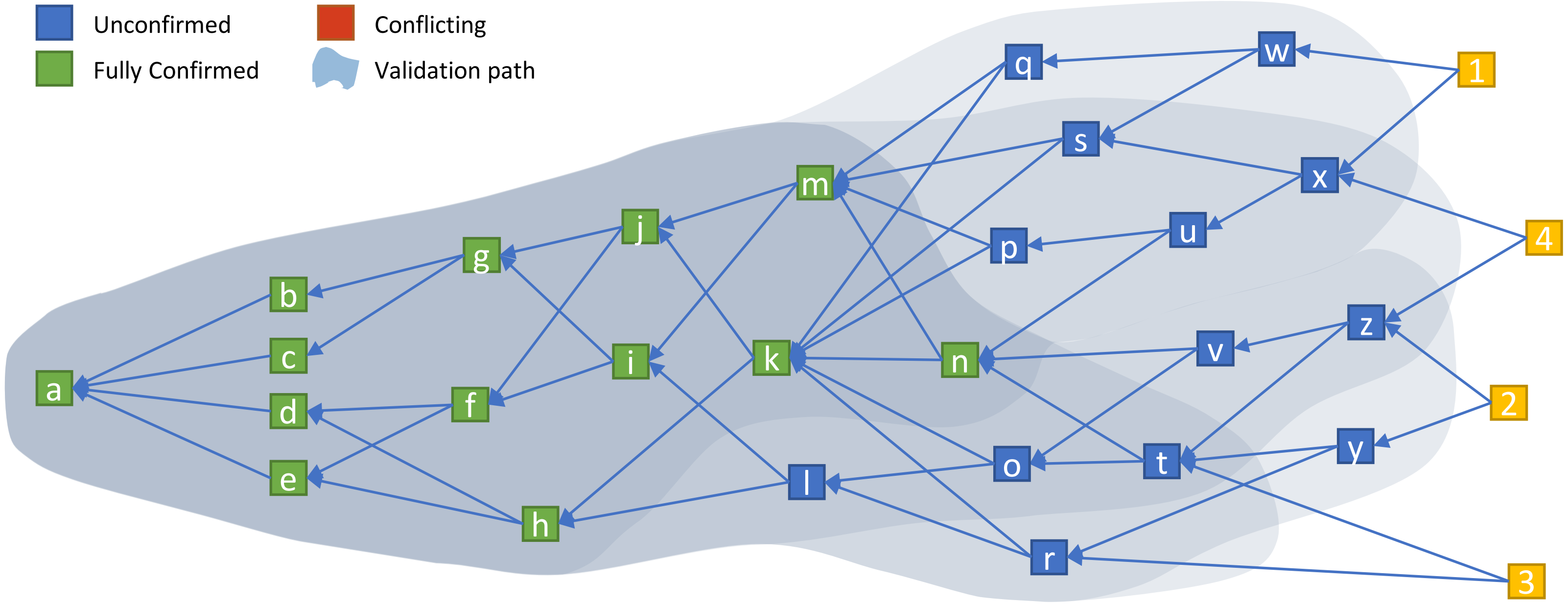
What did we learn?

- **Nobody needs to see and validate all transactions**. Every user just needs to select and validate two transactions and their parents. By doing so they are only validating a part of the tangle. As other users select and validate different tips and paths, a collaborative validation of the complete tangle emerges.
- After some time, once a transaction is deep enough in the tangle, a direct or indirect path from any of latest tips towards it exists. Such a transaction is considered fully confirmed and **is going to be re-validated and re-confirmed again by every single new transaction.** We can assume that it got confirmed by all users (and machines) and has high certainty.
- In order to check **for confirmation, a recipient only needs to check whether the transaction is directly or indirectly referenced all available tips yet** (or by a certain rate of them if a lower certainty, such as 80%, is accepted). No re-validation or similar is necessary. Note: There might be thousands of tips. Instead of checking the parents of each of them, it is possible to select a random sample and do a statistical evaluation.

Note that transaction “n” did not just get confirmed because we have less tips now. The next slide shows the same sample with more tips.

Confirmation Levels

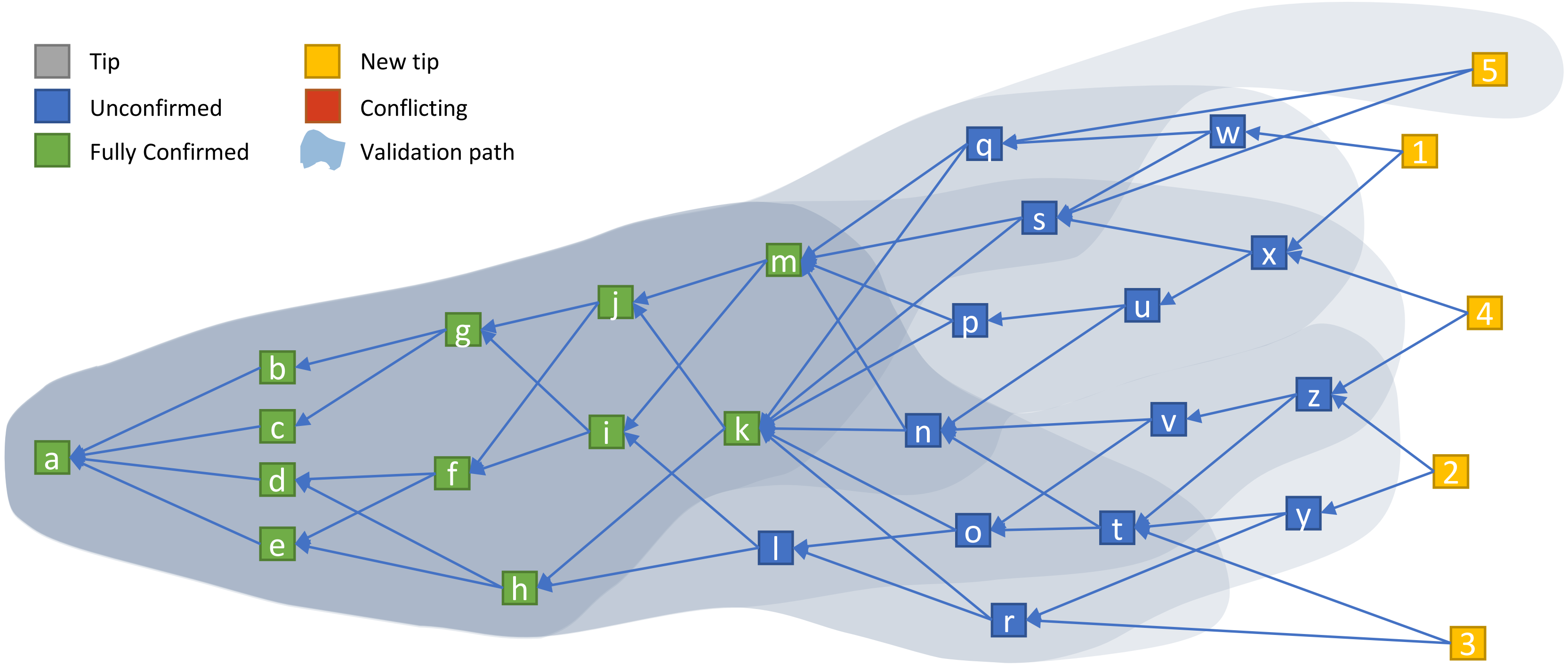
◼ Tip
🟦 Unconfirmed
🟩 Fully Confirmed
🟨 New tip
🟥 Conflicting
☁ Validation path



I added some more new tips to show an extended example. For each new tip its validation path is highlighted. By the coloring you can see well which transactions are validated by how many tips and their confirmation levels.

A merchant may choose a custom confirmation/certainty level. If transaction speed is more important than the value of the transaction (e.g. a micro transaction or zero value transaction), or if the sender is a friend, one may accept something such as a 75% confirmation level. At a 75% certainty level (3 out of 4 tips), the transactions “l”, “o” and “t” could be considered as confirmed, as well.

Propagation Delay

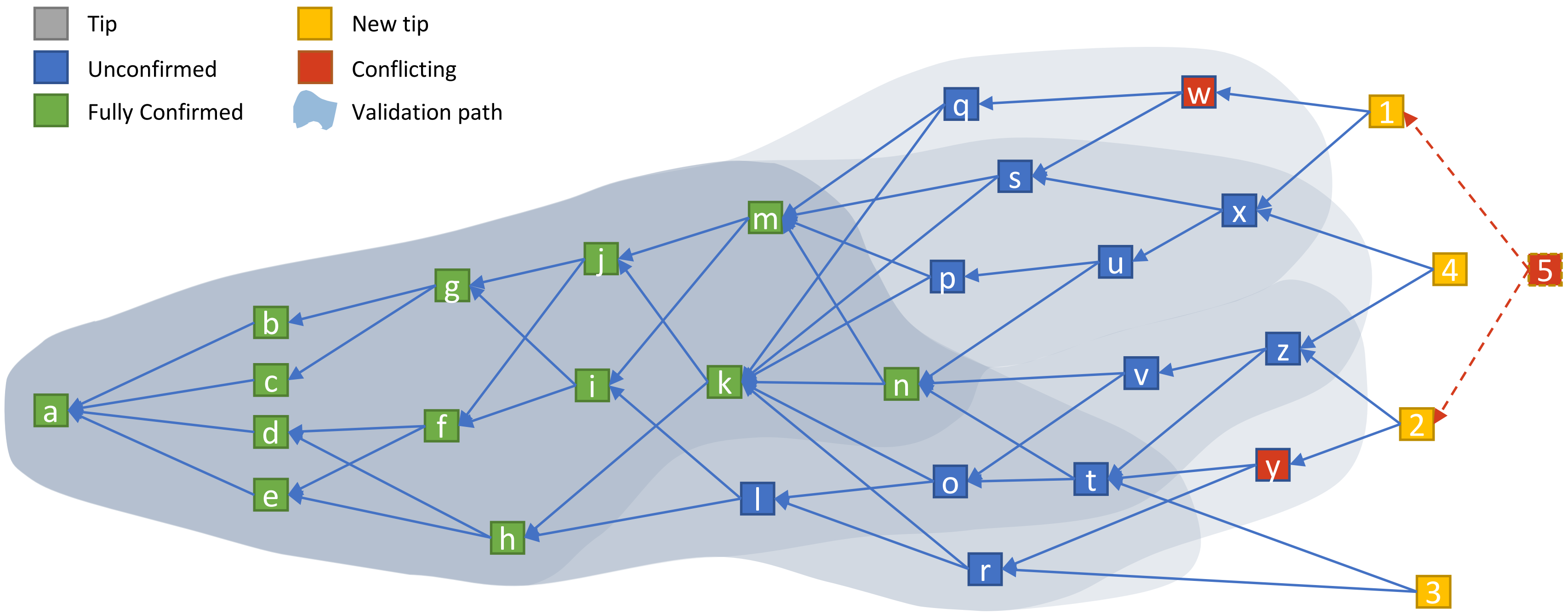


Theoretically it could happen that a slow transaction “5” pops up a bit later, due to slower PoW or due to propagation delays. Now that we know of transaction „5“, the transaction „n“ is not fully confirmed by all tips anymore. However, their confirmation certainty is still quite high with 4 out of 5 tips (in reality there would be thousands instead of 5 tips). Keep in mind, it’s all about a high probability of certainty – just as in blockchain technologies with its confirmations where each block on top increases the probability of certainty.

Please note that transaction “5” in this example is NOT flipping the transaction’s state from “confirmed” back to “unconfirmed”. It just changes the mathematically exact number of certainty (e.g. if there were 100 tips in total, from 100% to 99%). Once some subsequent transaction references (e.g.) transaction “1” and “5”, the transactions “n” is fully confirmed by all tips again. Such minor confirmation level variations will even less likely happen, the further transactions move into the tangle.

Please note that a confirmation/certainty level of a 100% might be hard to achieve anyways, as there could always be some troll tip around (e.g., referencing something useless or not following the protocol).

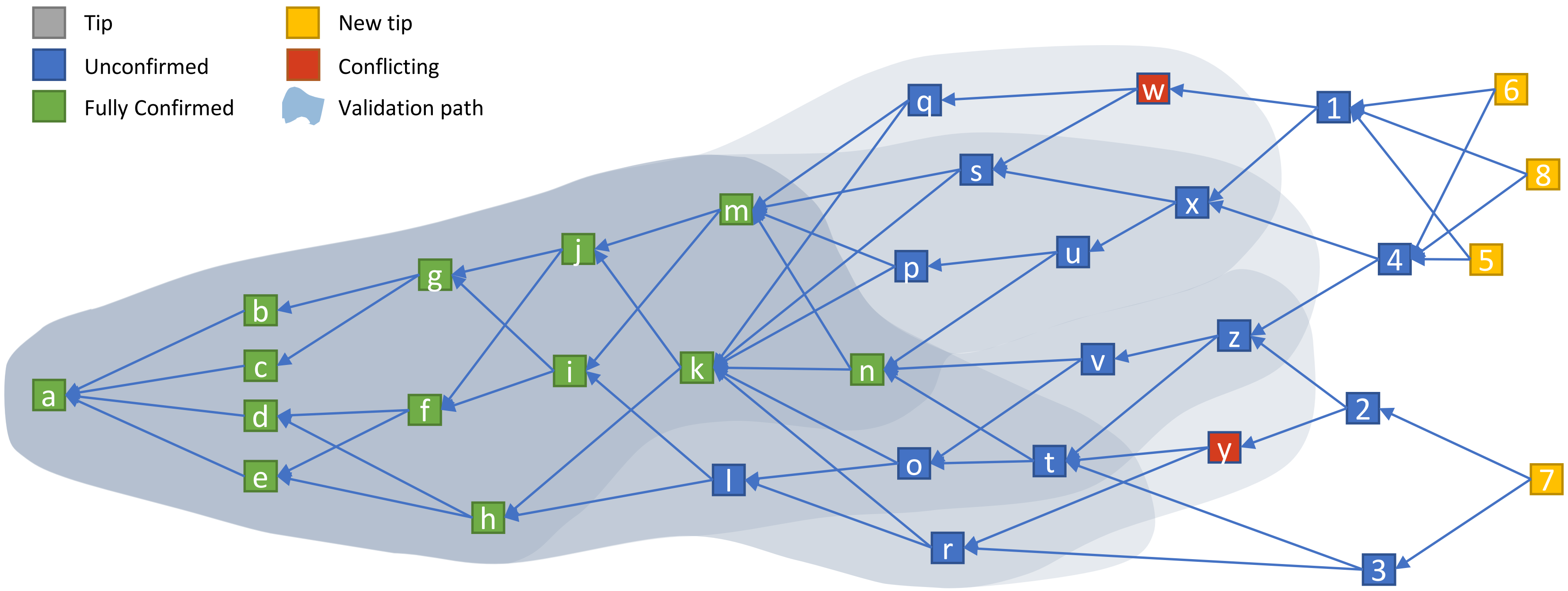
Double Spend



Imagine a situation where a user adds two conflicting transactions in different areas of the tangle (“w” and “y”). Subsequent users might only have one of these conflicting transactions in their validation path (based on their tip selection and maybe due to propagation delays). For example, the users attaching transactions “1” and “2” will not see the conflict and confirm their chosen tips. Hence, the double spend attempt got its first confirmations. However, sooner or later it must happen, that both conflicting transactions are in the path of validation of one transaction. For example, transaction “5” would see the conflict and not attach to the elected tips. Instead it would reselect tips until it found to not conflicting ones in order to be sure itself turns into a valid transaction.

Depending on the tip selection and tangle progress, it might happen that many more users attach their transactions behind “w” OR “y”, before the conflict becomes clear. Depending on where users attached most new transactions, either “w” or “y” will confirm at some point, while the other gets abandoned. All subsequent transactions attached to the abandoned one (as they couldn’t see the conflict coming) will also be abandoned. However, they are not lost but can be taken by anybody (but most likely the payment recipient) and reattached to the tangle for a new chance of confirmation. The PoW would need to be redone, but no fresh signatures from the sender are required.

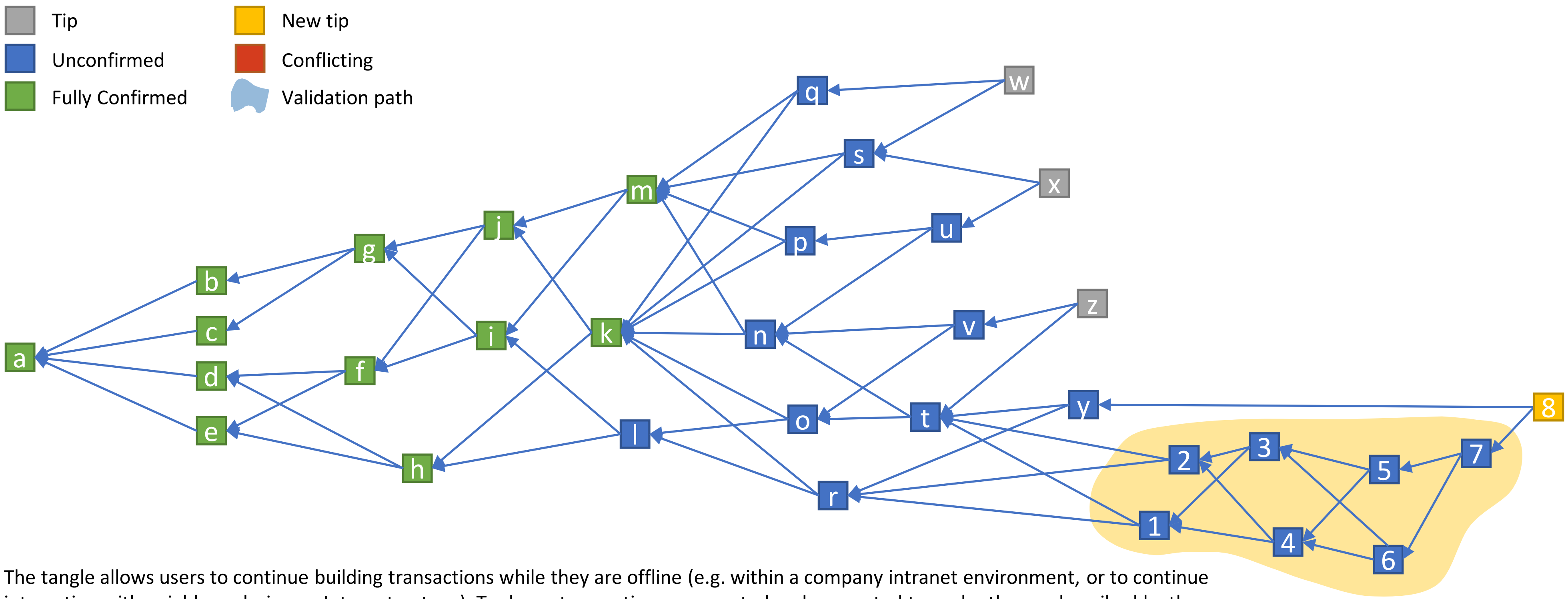
Double Spend Resolution



In the previous slide a user tried to connect a transaction “5” to the tips “1” and “2”. Due to a conflict it retried the tip selection, and decided to attach to tips “1” and “4”. Another user (or maybe the same one) chose tips “2” and “3” to attach transaction “7”. Some kind of branches emerged, but only one can survive, due to the double spend in “w” and “y”. Based on the random selection of tips (and the cumulative weight of the transactions), one of the two branches will receive more child transactions (respectively, weight) until the tangle turns into a state where it is not possible to attach legitimately to one of the segments anymore. In the sample above users could just continue attaching to transactions “5”, “6” and “8” but not to transaction “7” anymore. Hence, transactions “y”, “2”, “3” and “7” will never make it into a fully confirmed state.

As described in the previous slide, transactions “y”, “2”, “3” and “7” could be reattached to the tangle again. As long as they are (still) valid, they have a fresh chance of confirmation. Transactions “2”, “3” and “7” might become confirmed then, while transaction “y” will stay invalid.

Offline Tangle



The tangle allows users to continue building transactions while they are offline (e.g. within a company intranet environment, or to continue interacting with neighbors during an Internet outage). To do so, transactions are created and connected to each other as described by the protocol.

In the sample above, transactions “1” and “2” are the first offline ones. They are connected to the last known tips of the online tangle. Subsequent transactions attach as usual. Once a commit to the main tangle is desired (or possible, in case of an Internet outage), the offline sub tangle is finalized by creating transaction “8”, which is merging the offline tangle with a recent tip of the online tangle. Subsequently, transaction “8” turns into a legit tip and can be selected and validated by later online transactions. The next users attaching to transaction “8” online will include all offline transaction in their validation routine.

Please note, that offline transactions can only become fully confirmed once they made it into the main tangle like any other transaction, just as shown on the previous slides. If any transaction within the offline branch was conflicting with the main tangle, the transactions “1” to “8” would not become confirmed. Again it might take some subsequent transactions until the conflict is visible from all (or the majority) of the main tangle’s tips (as described in slide 8 “Double Spend”).