CONCORDIUM

Concordium White Paper

An introduction to the features of the Concordium Platform

Concordium is a privacy-focused, public and permissionless blockchain architecture. This paper describes the Concordium Platform along with a range of novel features that allow individuals, businesses and public institutions to use permissionless blockchain technology in a way that is private, trusted, scalable and compliant with regulations.

The Concordium Platform is designed to be fast, secure and cost-effective. Concordium's innovative identity layer provides on-chain identity, compliance-centric transactions and enhanced privacy for users, while also allowing for the de-anonymization of network participants. Concordium's two-layer consensus protocol consists of a Nakamoto-style consensus blockchain and a finality layer for fast confirmation of transactions. Concordium also enables interoperability and communication between Concordium and other blockchains. Concordium has a standards-based smart contract core with multi-language support. The Concordium Platform also features a transparent incentive structure with cost-effective transactions and predictable fees.

An Introduction to Concordium
Version 1.6, January 2023

Concordium: An Overview

Regulatory Compliance by Design

Concordium is designed to integrate with current financial and business systems that require knowledge of a user's identity. Through the development of unique protocol-level identity primitives, Concordium helps application developers, individuals and businesses build products that comply with local regulations, while retaining the benefits of a privacy-focused, public and permissionless blockchain.

By user, we mean any entity that holds an account on the Concordium Platform. These can be individuals or legal entities, such as businesses, and they require a valid form of identification to facilitate the off-chain identification process.

Both Privacy and Verification of the Identity of Users

Concordium's innovative identity layer provides a compliance-centric balance between anonymity and accountability. A user's identity is anonymous on-chain, however, this anonymity can be revoked and their real-world identity revealed in response to a valid request from a government authority via established legal channels. From the user's perspective, anonymity towards the general public is maintained and Concordium's identity layer can accommodate identity providers and anonymity revokers based in different jurisdictions around the world. As such, the Concordium Platform offers a global, multi-jurisdictional solution to the adoption of blockchain technologies across regulatory regimes.

Fast Transactions at Scale

The Concordium Platform is designed to be fast enough, in terms of transactions per second and the time it takes to finalize a transaction, to meet the needs of business applications on a global scale. This is a major development compared to previous generations of blockchain technology.

Provable and Fast Finality

Concordium has developed the first provably secure and fast finality layer to run on top of a Nakamoto-Style (NSC) blockchain. This means that a transaction on the Concordium Platform is confirmed and immutable within a short period of time. This is a major advantage over other NSC blockchains, where the finality of a block is only assumed after a large number of subsequent blocks have been produced.

Reliable Uptime

Concordium is designed for demanding business use cases with strict uptime requirements. Our two-layer consensus design ensures that the platform remains available and secure in the most adverse conditions. Blocks are correctly added to the longest chain when less than 50% of all stake is controlled by malicious parties, and we achieve significant speedups and efficiencies under normal conditions when less than 33% of all stake is controlled by malicious parties. Furthermore, the safety of our finality layer holds even under catastrophic failures to the network that causes messages to be delayed much longer than under normal conditions.

High Throughput for Global Scale

Many real-world business applications of blockchain technology have high throughput requirements. To meet these needs, Concordium is developing novel scalability mechanisms that operate in tandem with our consensus layer and help enhance the growth of the entire ecosystem without sacrificing security or decentralization.

A Standards-based Smart Contract Core with Multi-Language Support

Concordium's core on-chain language is WebAssembly (Wasm), a portable, well-defined assembly-like language. Wasm is an internet standard that is gaining a lot of traction in recent years and is already supported in the major web browsers. Many programming languages can already be compiled to Wasm, which potentially allows us to support a large range of smart contract languages. Concordium uses Rust as our first high-level smart contract language, a safe language that also allows for low-level resource control.

A Single Native Token That Is Easy to Model

The native token of the Concordium Platform is called CCD (short for ConCorDium) and has the symbol Ͼ. It is the medium of incentivization that ensures network participants are rewarded for their efforts. CCD can be used for a variety of purposes, including as payment for the execution of smart contracts and payment via transactions between users.

Cost-Effective Transactions

To accommodate businesses that operate at scale, transactions on the Concordium Platform are designed to be cost-effective as well as fast. Low costs are a function of our proof-of-stake and finalization along with incentive mechanisms that prevent excessive charging.

Know Transaction Costs Ahead of Time

Transaction costs need to be understood ahead of time in order to build sustainable business models. Concordium uses an innovative price stability mechanism to ensure that transaction costs are fixed in real-world fiat terms despite potential volatility in the price of CCD on the open market.

Transparent Token Economics and Incentives

The economy and incentive structure within the Concordium Platform is transparent and easy to understand. All parameters that control the distribution of rewards, the CCD growth rate, and the pricing of transactions are publicly available. The Concordium Foundation will actively monitor and manage the health of the Concordium economy.

Responsible Governance on the Road to Decentralization

While the Concordium Foundation will act as a guarantor that the central principles of the Concordium Platform are adhered to, including privacy with accountability, key functions will be delegated to the Governance Committee. Over time the Governance Committee will evolve to have more responsibilities and governance will be decentralized across network participants.

Open Source

The Concordium Platform has been developed in a closed environment, but before the launch of Mainnet, the codebase for all central components of the Concordium Platform has been made open-source at https://github.com/Concordium.

Design Overview

The Concordium protocols can be divided into the following layers:

The **network layer** consists of a **peer-to-peer layer** and a **catchup layer.** The peer-to-peer layer consists of a public, permissionless, high-speed protocol for broadcasting messages to all available nodes. The catchup layer sits between the peer-to-peer layer and the consensus layer and ensures that nodes receive all relevant messages.

The **consensus layer** ensures agreement on all transactions and their order in the ledger. The consensus layer has at its lowest level a **proof-of-stake Nakamoto-style consensus blockchain** that uses the longest chain rule. On top of the NSC blockchain is a fast **Byzantine fault tolerance (BFT) finality layer**. In future updates, Concordium's consensus layer will be augmented to another layer that improves scalability even further.

Accounts and identities define how users' identities are processed and used during the creation of new accounts. It specifies how identity providers and anonymity revokers interact with users to create identity objects and revoke the anonymity of users if validly ordered by a qualified authority.

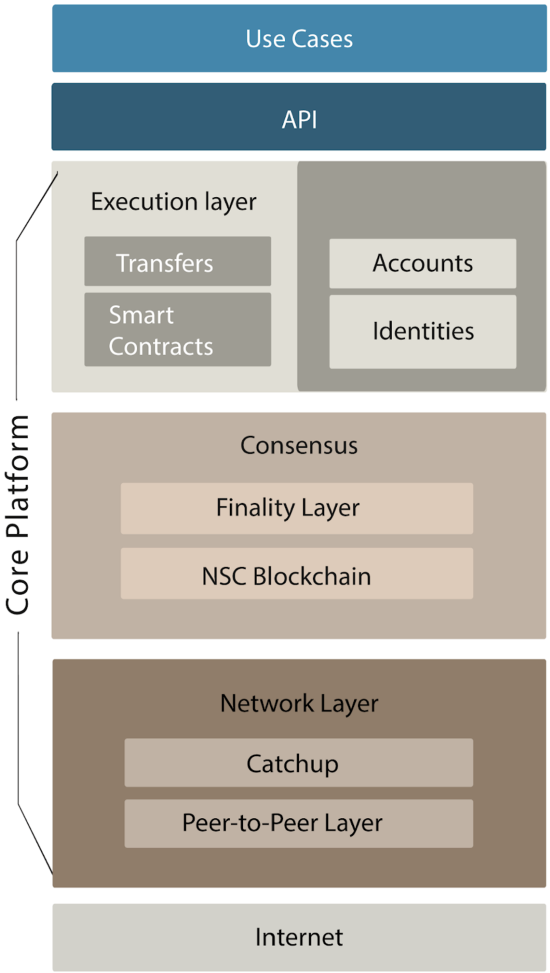

Figure 1:Overview of the Concordium stack.

The **execution layer** allows users to interact with the platform. Using the **API**, users can submit transactions, including, for example, transfers between user accounts and deployment and execution of smart contracts.

Network Layer

The Concordium Platform is a distributed system. It consists of multiple nodes which maintain the blockchain by baking and finalizing blocks. The network layer establishes and manages the communication between nodes.

The network layer provides an abstract interface for communication such that protocols sitting on top of it do not have to know details of the actual communication protocol. For example, the interface for the consensus layer is a broadcast channel that allows parties to send messages to all other parties.

The network layer consists of the peer-to-peer layer that manages communication and the catchup layer that ensures all nodes, including those having been temporarily offline, receive all necessary messages.

Peer-to-Peer Layer

The peer-to-peer layer establishes a node's connection with peers and manages bi-directional communication between them. Its role can be divided into 3 parts:

1. To set up and maintain a node's presence on the network. This includes finding and managing peer nodes and collecting information about the network to ensure optimal peer-lists.
2. To manage one-to-one communication with each peer.
3. To broadcast messages, including transactions, blocks and finalization messages, to all nodes in the network.

When a node starts up, the peer-to-peer layer is the first module that boots up. The node resolves a DNS record, decodes the list of bootstrap servers, connects and receives a list of peers. Initially, the bootstrap servers will be managed by the Concordium Foundation.

All communication on the peer-to-peer layer is done through network messages. For efficiency, network messages are serialized using FlatBuffer which enables fast implementation in various programming languages.

The peer-to-peer layer protocol is heavily secured against wiretapping by making use of the Verifiable Noise Protocol [Per16] and using formally verified encryption implementations. This enables a very robust guarantee against wiretapping and replay attacks on the peer-to-peer layer.

Any node can initiate a broadcast of a message, *m*, by sending *m* to its peers. When a peer receives *m* for the first time, it forwards *m* to all its peers. This ensures that the message propagates through the whole network. To avoid resending the same message multiple times, nodes buffer hashes of previously broadcast messages. A received message is then only forwarded to all peers if it is not already in the buffer.

Catchup

The peer-to-peer layer allows sending messages to all nodes that are online when the message is sent. The catchup layer ensures that nodes that were offline when the message was sent, or that missed the message for other reasons, also receive the message.

There are two types of messages that are handled differently by the catchup layer. The first type is messages where nodes can determine by themselves that they missed the message. This includes blocks sent in the consensus layer. For example, when a node receives a block with a parent-pointer to an unknown block, this shows that the message containing the parent block is missing. For such messages, the catchup layer uses a pull mechanism, i.e. nodes pull messages from their peers by explicitly requesting them.

The second type of messages is those that cannot easily be recognized as missing. For instance, this is the case for messages sent between finalizers in our finality layer. For these messages, the catchup layer uses a push mechanism in which the sender periodically resends these messages as long as they are relevant. In this case, finalization messages for a particular block are relevant and replayed until the finalization for this block is complete.

In the future, Concordium plans to add support for fast catch-up based on the following observation. In blockchain networks with a finalization layer like Concordium, a node only needs to validate the verification keys of the latest finalization committee in order to verify a claimed latest block. Having such verification keys, the node can simply verify if the claimed block has been signed by the supermajority of them. To validate the verification keys of the latest committee, one can now verify a chain of committee handover messages each certifying the verification keys of the next elected finalization committee. Assuming that there has been no intermediate committee with a dishonest supermajority, such a chain of committee signatures would suffice to validate the latest committee.

Consensus Layer

A major innovation of our design is the two-layer consensus approach, which combines a Nakamoto-style consensus blockchain with a novel finalization method, providing fast finality.

Proof of Stake and Delegation

The Concordium Platform uses a proof of stake (PoS) mechanism to ensure resource-efficient operation of the network along with enhanced security among participants. Users that hold CCD in their account can either become a baker, i.e., stake some of their CCD and run their own node, or delegate their CCD to existing bakers. A delegator has two options: 1) they choose a specific baker and add their stake to that baker's pool—thus increasing that baker's chance of winning the lottery to bake a block—and share some of the rewards; 2) they choose Concordium's novel passive delegation feature, which provides the delegator with the rewards from sharing their stake between all baker pools, thus reducing the risk of delegating to a single baker that performs poorly, but provides fewer rewards than delegating to a single baker.

Concordium's Two-Layer Consensus Mechanism

Nakamoto-style consensus

We define Nakamoto-style consensus (NSC) blockchains as systems where parties participate in a form of a lottery to win the right to append blocks to the chain. The probability that a certain party wins the lottery depends on how much of the required resource they have, for example, computing power for proof of work systems or stake for PoS systems.

In NSC blockchains it is possible that, due to network delays, one party adds a new block without having received information about the previous block. Furthermore, malicious parties can purposefully ignore existing blocks. Both of these situations result in a fork in the chain, turning the blockchain into a tree. One way to deal with this issue is for parties to have a chain-selection rule, such as the longest-chain rule, which determines which chain in the tree parties consider as their current chain and where new blocks should be added. The chain selected by any given party can change over time, causing rollbacks and invalidating transactions on the previously selected chain. Since very long rollbacks are unlikely, blocks can be considered 'final' when there are 'sufficiently many' blocks after it in the chain.

NSC blockchains are secure when corruption is below ½. In PoS systems this threshold refers to the fraction of the stake controlled by malicious parties.

Committee-based Byzantine fault-tolerant consensus

Aiming to provide an alternative, there are committee-based Byzantine fault-tolerant (CBFT) consensus designs such as those employed by Tendermint [Kwo14] and Algorand [CM19]. They provide immediate finality in that every block included in the chain can be considered final. This mechanism offers better consistency when compared with NSC blockchains, but this comes at a cost. Namely, CBFT consensus designs only tolerate corruption below ⅓ of the stake.

Our two-layer approach

Concordium's finality layer can be added on top of NSC blockchains [DMMNT20]. Our finality layer allows us to dynamically 'checkpoint' the blockchain by using Byzantine agreement to identify and then mark common blocks in the chains of honest users as final. This two-layer approach provides the best of both worlds. In particular, we achieve the following:

- As long as corruption is below ⅓, our finality layer declares blocks as final faster than the finality rule in a pure NSC blockchain, which is required to wait for 'sufficiently many' blocks.
- When corruption is between ⅓ and ½, we can still rely on our NSC blockchain and obtain the same guarantees as a pure NSC blockchain. Note that pure CBFT designs fail completely under these conditions.

Concordium's Nakamoto-Style Proof-of-Stake Blockchain

Blockchain protocol

While Concordium is currently undertaking research to improve the efficiency and security of NSC blockchains [KMM+21], our proof-of-stake mechanism is a simplified variation of Ouroboros Praos [DGKR18]. The simplification is made possible due to our innovative finality layer, which prevents long-range attacks.

The party that participates in the production of blocks is called a *baker*. Time is divided into equally sized periods called *epochs* each consisting of the same number of *slots*. Currently, an epoch is set to one hour and a slot is 250 milliseconds. In every slot, each baker checks locally whether they won the lottery. To this end, every baker has a secret key for a verifiable random function (VRF). This allows the baker to evaluate the function on given inputs such that other parties, who do not know the secret key, can verify that a certain output was computed correctly without being able to predict outputs themselves.

To check whether a given baker has won in a given slot, it takes the values *slot* and a *nonce* as inputs to the VRF and computes a random value r. The *nonce* is a random value that is updated for each epoch to prevent parties from predicting too far into the future when they will win. The party wins if the value r is below a threshold T, which depends on the party's relative stake α and a common difficulty parameter *f*. The winning probability is Prob(Baker with relative stake α wins) = 1 - (1 - *f*)^α.

The winning probability is roughly proportional to α, and higher difficulty parameters increase the winning probability for all parties.

A winning party then extends the current longest chain by a fresh block. Since parties compute their VRFs independently, there can be zero or multiple winners for a given slot. The slot time and the difficulty parameter f are set to ensure that blocks are produced with the expected blocktime, and to reduce the likelihood of having multiple winners.

The lottery uses a fixed relative stake for each baker per pay day, which is set to be 24 hours. Rewards for baking are distributed at the end of the pay day. Under normal network conditions, the stake of a baker for a pay day is determined one hour (one epoch) before the end of the previous pay day.

Obtained guarantees

Running our NSC blockchain protocol described above, participating bakers grow a tree with the genesis block at its root. The consistency attribute we obtain is called the common-prefix property. It says that if the majority of the stake is controlled by honest parties who follow the protocol, the longest chains in the view of all honest parties have a long common-prefix. That is, all honest parties agree on all blocks except for the freshest blocks. We also obtain the properties of chain quality, such that a sufficient fraction of blocks are generated by honest bakers, and chain growth, such that the longest chain grows at a sufficiently high rate. Thus the protocol is both safe and live. These properties of the Concordium protocol have been formally verified in [TS21].

Concordium's Finality Layer

Overview and guarantees

As the tree of blocks described above grows, a finalization committee is responsible for periodically marking blocks as final. Bakers only extend the chain past the last finalized block and finalized blocks are never rolled back.

Finalization committees

Finalization is run by a subset of the bakers in the **finalization committee** whose members we call **finalizers**. For finalization to work properly, we need honest finalizers to hold more than ⅔ of the total stake, including delegated stake. It is therefore important to select the finalization committee such that sufficient honest stake participates in finalization. On the other hand, committees that are too large will slow down finalization considerably. We balance these two conditions by allowing all bakers that hold a minimum required fraction of stake to be included in the finalization committee and act as finalizers. Currently, finalizers are required to hold at least 0.1% of the total stake, which ensures that there will be at most 1,000 finalizers in the committee and that all nodes with substantial stake can participate.

Sketch of our finalization protocol

Recall that our NSC blockchain produces a growing tree of blocks. The finalization committee is repeatedly finalizing blocks in this tree. In each iteration, finalizers run a protocol to agree on a unique block at a given depth d, i.e., with distance d from the genesis block. Finalization for a given d proceeds as follows. Each finalizer waits until its chain has reached depth $d + \Delta$, where Δ is a parameter that is set to 0 at genesis. The finalizer then votes on the block it sees at depth d on its chain using the Concordium CBFT consensus protocol. This protocol is designed such that it succeeds if all finalizers vote for the same block, otherwise it might fail. If the consensus protocol is successful, the block it outputs is defined to be final. If the consensus protocol fails, the finalizers iteratively retry until it succeeds. In every retry, the value of Δ is doubled (or set to 1 if it was 0) and the finalizers wait until they are at depth $d + \Delta$. They then vote for the block they see at depth d on their chain. Eventually Δ will be large enough that the block at depth d is in the common-prefix of all finalizers. In that case, all finalizers vote for the same block, and the consensus protocol will succeed. Increasing the offset Δ exponentially can be seen as a binary search for the common prefix. Note that honest parties typically only deviate for a few blocks, so the algorithm will usually succeed with small Δ. After successful finalization, the finalizers produce signatures on the finalized block, which count as a finalization proof. The finalization proof will be part of a subsequent block. Finalization is then repeated for a larger depth d, and the value Δ is divided by 2 (or decreased from 1 to 0). Decreasing Δ ensures that a good value is found over time.

Our finality layer works if there is *some* common-prefix; it does not need to know how long it is. The rationale behind this is two-fold. It gives responsive finality, which means whenever blocks are included in the common prefix of all finalizers and subsequently agreed upon quickly, we also finalize quickly. Furthermore, not relying on a fixed bound for the common-prefix length makes the finality layer work as a hedge against catastrophic events that cause long forks (e.g., partitions of the internet).

Obtained guarantees

As proven by Dinsdale-Young et al. [DMMNT20], we obtain the following guarantees from our finality layer:

- **Chain-Forming:** finalized blocks form a chain;
- **Agreement:** all parties agree on the finalized blocks;
- **Updated:** the last finalized block does not fall too far behind the last block in the underlying blockchain;
- **⅓-Support:** all finalized blocks are ‘supported’ by honest parties holding at least ⅓ of the total stake. This means that honest parties had these blocks on their chains before finalization and they are not required to adopt a new chain after finalization, limiting potential rollbacks.

To further ensure the reliability of our platform we are investigating formal verification methods to formally prove the security of our finality layer [DSTT19].

Scalability

Our current two-layer consensus protocol consists of a Nakamoto-style consensus and a finality layer that enables high transaction throughput and fast confirmations. Concordium is implementing a new highly-parallelizable consensus mechanism that offers more built-in scaling advantages. To address scalability needs even further, an in-depth research phase is currently underway, investigating a number of different approaches. We discuss the most important ones below.

Sharding

The main goal of sharding is to overcome scalability issues. Without sharding, every node in the network has to process all transactions and execute all smart contracts. The basic idea of sharding is to parallelize execution by dividing the network into smaller components, called shards. Nodes are then assigned to different shards with separate account balances. Each shard essentially corresponds to a separate blockchain that can be running almost independently of the other shards. This means that transactions on one shard are only processed by the nodes on that shard and, consequently, more transactions can be processed overall.

Overall sharding architecture. Concordium is investigating a two-level sharded design with a very robust control chain and light weight shards. Our sharding mechanism has been described by David et al. [DMMNT22].

The control chain manages shards, provides a finalization service to the shards and gives a vehicle for cross-shard transactions. Each shard runs an individual blockchain and uses the control chain to coordinate the individual shards.

As shards allow for efficient reading of only the parts of this global ledger needed for a given application, this architecture makes it possible to create shards for specific purposes including private shards as described below.

Obtaining security and efficiency. For optimal efficiency, there should be many shards, each run by a small committee. However, the risk associated with sampling only a small number of nodes per shard is that a large fraction of them might be corrupted. This means the usual consensus protocols, which require a (super)-majority of honest parties to be secure, cannot be used.

The important insight allowing us to solve this issue is that security consists of two parts: safety and liveness. Safety means the system does not make mistakes. Liveness means the system does not come to a halt. Furthermore, many consensus protocols can be adapted, such that one can trade liveness for safety. For example, one can tolerate <80% of corruption for safety at the cost that liveness is only guaranteed for corruption <10% which is in contrast to the usual <33% for both properties.

We will utilize this insight as follows. The control chain consists of many nodes with broad decentralization. The shards run with few nodes per shard, tolerating relatively high corruption for safety. This gives us scalability and safety but with limited liveness in the shard. To improve the latter, the control chain monitors the shards and if any come to a halt because of too many corrupted nodes, it resamples the set of nodes running that shard. This re-establishes liveness and with safety and liveness in place, we have security.

Initially, Concordium will run the same consensus algorithm on the shards as on the control chain. However, different shards could use different algorithms, and Concordium will research the usefulness of supporting more consensus algorithms to meet the specific requirements on different shards.

Intershard signaling. Intershard signaling allows transactions between shards and communication of smart contracts on different shards. When a block finalizes on a shard, it contains a list of outgoing messages for other shards. The nodes of the sending shard sign the list of outgoing messages. The nodes in the receiving shard can obtain the list of nodes running on the sending shard together with their public keys from the control chain, which allows them to verify the signed messages from the sending shard. Once a message is verified, it is executed on the receiving shard.

Private shards. The sharding mechanism would also allow for private shards. In a private shard, the control chain cannot see the transactions on the shard, it only provides Finalization-as-a-Service (FaaS) and coordination to relaunch deadlocked shards. The private shard can ultimately run its own consensus algorithm and use its own identity providers and anonymity revokers. Private shards provide a cheap way for an individual, country or corporation to launch their own blockchain while having the benefit of the strong finalization service provided by the control chain.

Layer 2 Scaling

Layer 2 scaling techniques propose another approach to help scale the blockchain. These techniques—commonly referred to as “off-chain” scaling solutions—serve as a separate blockchain built on top of a base network like Concordium. A layer 2 protocol increases transaction speed by processing transactions off the chain while still ensuring similar security and decentralization guarantees as the base network.

The main layer 2 solutions are zero-knowledge (zk-) rollups and optimistic rollups. Zk-rollups are smart contracts that condense hundreds of transactions executed off-chain into one on-chain transaction and submit a short cryptographic proof on-chain that ensures the validity of the executed transactions. By storing only a small fraction of total data on-chain rather than all transaction data, zk-rollups make the base network cheaper and faster. As opposed to ZK-Rollups, optimistic rollups do not rely on cryptographic proofs. Instead, they are “optimistic” and assume transactions are valid. Only if the result of a rollup transaction gets challenged via a fraud-proof does the protocol correctly updates the rollup’s state and penalizes the responsible rollup sender who included the invalid transactions. Concordium is investigating rollup scaling solutions as a support for the main chain.

Identity Layer

This section describes Concordium’s innovative identity layer that allows users to create a verifiable identity off-chain to ease compliance with relevant regulations, while also allowing that identity to be represented on-chain in a way that protects the user’s privacy. Concordium’s identity layer has been proven secure in [DGKOS21].

Previous blockchains have chosen extreme balances between user privacy and accountability. Some blockchains allow fully anonymous transactions without any accountability, making them vulnerable to illegal activity. Equally troubling is that while some blockchains do not provide true anonymity for transactions, allowing for transactions and accounts to be tracked, they offer no systematic way to discover the real-world identity of suspicious users.

Concordium offers a workable solution by providing transactional privacy for users, along with a mechanism that allows accountability to local regulations. This means that transactions are processed without exposing the identity of the sender or receiver. In case of shielded transfers (explained later), the sender and receiver will also be the only parties that can see the actual amount of a transaction. On the other hand, where a suspicious transaction or set of transactions have been detected, the real-world identity of the user can be obtained by qualified authorities with the help of anonymity revokers and identity providers. Moreover, if a specific real-world identity is the subject of an authorized investigation, anonymity revokers and identity providers can help trace all accounts and transactions of that user.

Entities in the Identity Layer

This section provides an overview of the entities involved in the identity layer.

Account holders

Account holders are individuals or companies who hold accounts on the blockchain. Before opening accounts on chain, account holders need to register with an identity provider.

Identity providers

An **identity provider** is a person or organization that performs off-chain identification of users. For each identity issued for a user, the identity provider stores a record off-chain called an **identity object**, and the user gets a corresponding private part of the identity object, called **user identity certificate**[1], which is known only to the user.

The primary functions of an identity provider are to:

- Verify the identity of users;
- Issue user identity certificates to users;
- Create and store identity objects and relevant attributes for record-keeping purposes; and
- Participate in the anonymity revocation process.

Information about the organizations that act as identity providers, such as their name, location or public key, is found in an on-chain registry. Initially, the registration of identity providers will be managed by the Concordium Foundation. Users must go through the identification process with a registered identity provider in order to open and operate an account on the Concordium Platform.

Anonymity revokers

An **anonymity revoker** is a person or organization that is trusted by the Concordium Platform to help identify a user that owns an account should the need arise. Initially, anonymity revokers will be appointed by the Concordium Foundation.

All accounts on the Concordium Platform are associated with a real-world identity, which is linked to an identity object stored by an identity provider. Identity objects are also linked to a set of anonymity revokers. Anonymity revokers play a critical role in revealing the real-world identity of a suspicious user by decrypting the **unique user identifier** that is stored on-chain for each account. When a unique user identifier has been decrypted following service of an official order (as described below), it can be combined with information stored by the relevant identity provider to allow the qualified authorities to obtain the real-world identity of the user.

Processes

In this section we describe the processes related to the identity layer.

Account holder registration

Before an individual or entity can use the Concordium Platform, their real-world identity must be verified and recorded by an identity provider. To that end, the user---with the help of a government-issued identity document (e.g., passport)---must complete the process of creating user identity information via a purpose-built wallet or app.

The identity provider verifies as part of this process that the attributes in the user identity information are valid for the user. If the verification is successful, the identity provider stores a record and provides the user with an identity certificate that can be used for creating accounts on the Concordium Platform. These certificates are valid for a given period and users can obtain new certificates in connection with updated identity verification by an identity provider.

[1] User identity certificates should not be confused with other types of certificates such as X.509, which are public certificates. The user identity certificate is private to the user and never shown on the chain.

Creating accounts on the Concordium platform

Once a user has acquired a user identity certificate from an identity provider, they can create accounts on the Concordium Platform. This is typically done using an app or wallet that guides users through the account creation process.

The private account keys are stored by the user, whereas public **account creation information** is published on the blockchain. The latter contains public account keys, and information about the identity provider and anonymity revokers. While this allows the relevant anonymity revokers and the identity provider, when working together, to link the account to the user, the account creation information does not allow other parties or any single party to identify the user. Further, accounts created with the same user identity certificate cannot be publicly linked.

Multi-user accounts on the Concordium platform

The Concordium Platform also allows users to create jointly owned accounts. For example, three users can have a joint account where two of them are needed to authorize a transaction. The total number of users and the authorization threshold can be configured freely by the users. To create a multi-user account, one user creates a normal account and the other users generate account credentials that can then be added to the account by the initial owner. All added credentials contain identity information on the users, which, similar to a normal account, allows the relevant anonymity revokers and the identity provider, when working together, to link the account to the users.

Anonymity revocation

The identity of a user can only be revealed to a qualified authority as part of a valid legal process. A **qualified authority** is a governmental body that has the authority to act in a relevant jurisdiction. For example, a local police force, a local court or an investigatory division of a local authority that regulates financial conduct may have authority to act in their relevant jurisdictions. These authorities are qualified to begin the process of revoking the anonymity of a user when they proceed through established legal channels and make a formal request. The outcome of such a request is likely to be that a qualified authority obtains an **official order**, which may be in the form of a warrant, court order, or similar instrument. Only after a qualified authority validly serves an official order upon the relevant anonymity revokers and identity provider, can the real-world identity of a user be revealed and only to the extent set out in the order.

Concordium’s identity layer is flexible and sensitive to the evolving nature of financial regulation and its impact on the blockchain space. Where new legislation or rules emerge, for instance the application of the so-called “travel rule” to blockchain transactions, the Concordium identity framework offers a compliance-centric solution that can be tailored to specific business needs.

After the authorities have identified an on-chain transaction or account they would like to investigate in order to reveal the real-world identity of a user, the following process must be followed.

- The qualified authority must identify the anonymity revokers and identity provider associated with the account under investigation and present them with an official order. This information is available on-chain as part of the account creation data.
- Per the terms of the official order, the anonymity revokers will extract parts of[2] the unique user identifier for the user by inspecting and decrypting the available on-chain data.

[2] More precisely the unique user identifier is “secret shared” and after decryption each anonymity revoker can return its share of the identifier.

- The qualified authority can now combine the parts received from the anonymity revokers to reconstruct the unique user identifier.
- With this unique user identifier, the qualified authority can work with the relevant identity provider to retrieve the real-world identity of the user.

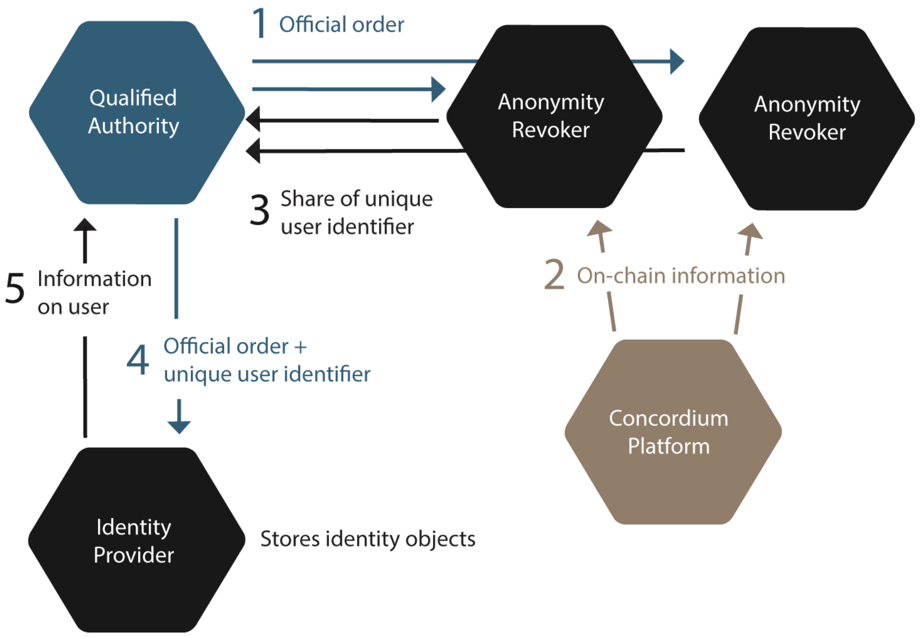

Figure 2: The anonymity revocation process.

If required and per an official order, all accounts and transactions related to a specific user can be uncovered in a similar process.

It is important to note that this process intends to dovetail into contemporary legal systems that have established checks, balances and controls to prevent overreach by qualified authorities. It also provides that a user can create accounts for which neither identity providers nor anonymity revokers acting alone can reveal an on-chain user's real-world identity on their own.

Proving Statements On Identity Attributes

Concordium wallet allows account holders to store and manage their attributes that have been issued by identity providers. An account holder by being in control of their wallet can decide in a self-sovereign manner how and when to provide their attribute information to applications. When an application needs to verify some statements of attributes, the account holder can generate a zero-knowledge proof that attests to the truth of the requested statement. This allows account holders to convince the application that they meet verification requirements without their personal data, apart from the statement, ever being collected.

The proof system provides the following functionality:

- (in)equality of attribute with a public value
- Set (non-)membership of attributes in a public set
- Range proofs, e.g., attribute <= public value
- AND combination of the above

For example, one can prove that "I am older than 18 and I am resident within the EU".

All proofs are created and verified off-chain. Moreover, in a future release, Concordium plans to add support for non-transferable proofs by which a prover can generate a proof to an intended receiver in a way that only the specified receiver can be convinced about the validity of the prover's claim. Such proofs can provide plausible deniability for the prover and ensure that the verifier cannot forward the proof to a third party.

Verifiable Credentials

In future updates, Concordium's identity layer will be augmented to support Verifiable Credentials. A verifiable credential system includes three roles: Issuer, Holder, and Verifier. An issuer attests some facts about holders in the form of verifiable credentials. For example, a book club can attest memberships or an organization can attest the completion of a professional program by issuing a diploma. To ensure decentralization, anyone—under some requirement—has the authority to become an issuer.

When needed, a holder can use the credential to prove arbitrary statements about their attested claims to the verifier in zero-knowledge, and a verifier will verify the claim by checking the validity of the credential proof provided by the holder. The trust level of the statement depends on the trust level of the issuer. Anyone with an account on the Concordium platform can become a holder. To verify a credential proof, one only needs read access to the Concordium blockchain.

Execution Layer

Users interact with the Concordium Platform via different types of transactions. Once transactions are submitted, they are added to the transaction pool. Bakers check the validity of transactions and include valid transactions into the next block. Transaction validity and the manner in which transactions are handled is dependent on the type of transaction. We discuss the most important types below.

All transactions have timeouts. The **timeout** is set by the creator of the transaction and a transaction cannot be executed after its timeout time has expired. This prevents the situation where an old payment can suddenly execute days later.

Transaction Types

Transfers between accounts

A user opens a new account by publishing an **account-opening transaction**. This transaction contains account creation information (see the above section, *Identity Layer*).

Plain transfers

The most basic type of transaction is a **plain transfer** of *x* CCD from *Account A* to *Account B*. Such a transaction is valid if the public balance of *Account A* is at least *x* + *transaction fee*. The effect of the transaction is that the public balance of *Account A* is reduced by *x* + *transaction fee* and the public balance of *Account B* is increased by *x*. Both accounts and the value *x* are publicly visible for this type of transaction.

Shielded transfers

The Concordium Platform supports **shielded transfers**. To allow for shielded transfers, accounts have a shielded **balance** in addition to their public balance. A shielded transfer has the same functionality as a plain transfer except that it operates on the shielded balances and the transferred amount is hidden and only known to the sender and receiver. To ensure that the sender has a sufficient shielded balance, the transaction contains a zero-knowledge proof that allows everyone to verify that the amount in the transfer does not exceed the sender's shielded balance, without revealing any of these values.

Authorities are able to reveal shielded amounts in a process similar to that used for anonymity revocation.

Anonymous transfers with accountability

While shielded transfers hide the amount of CCD it is possible to see which accounts are involved in the transfer. Concordium is investigating how to add an additional level of privacy with **anonymous transfers**. An anonymous transfer has the same functionality as a shielded transfer with the addition that the sender and receiver of a transaction cannot be linked.

Note that the anonymity revocation process still allows authorities to obtain information about the transaction hidden in an anonymous transfer.

Register data

Concordium also allows small strings to be put on the blockchain, e.g., to register the hash of a document. This protects the integrity of the document (since changes can be detected, as the hash will not match any more) and also serves as a timestamp.

Smart contract-related transactions

The life of a smart contract starts with the deployment of the smart contract code. Any user can deploy smart contract code using a special transaction and can get an instance of a deployed smart contract using an **initialization transaction**. Each initialized smart contract is associated with an account. An instance of a smart contract can receive inputs in the form of **input transactions**. These transactions specify an amount of ENERGY that is allowed for the execution of the input. The order in which different inputs are executed depends on the order in which the input transactions are added to blocks.

Consensus-related transactions

Users can become bakers by publishing a special transaction. The newly created baker is associated with one of the user's accounts, which is used for holding the staked amount of CCD and for receiving rewards. Furthermore, there are transactions to change the stake of the baker, to open or close the pool to new delegators or any delegators, to change whether the rewards are restaked, and to deregister as a baker.

Users can also delegate stake to bakers, either to a specific baker pool or using the passive delegation feature. Transactions for changing the stake, changing the delegation target and changing whether the rewards are restaked are also available.

Smart Contract Languages

WebAssembly low-level language

Concordium's core on-chain language is WebAssembly (Wasm), a portable, well-defined assembly-like language. Wasm is an internet standard that is gaining a lot of traction in recent years and is already supported in the major web browsers.

Many programming languages can already be compiled to Wasm, which potentially enables support of a large range of smart contract languages. Wasm allows for low-level control of the on-chain code, which helps with optimizations when adding support for cryptography in smart contracts.

Among permissionless blockchain platforms, there are very few common standards for smart contracts, however, Wasm is one of the few that is seeing adoption by multiple platforms.

Rust high-level language

Concordium plans to support a number of smart contract languages and has chosen Rust as the first high-level smart contract language. The Rust ecosystem is quite friendly, with good documentation and good support for WebAssembly.

Rust is a safe language, but it also allows for low-level resource control. This can help reduce the cost of contracts and makes it very well-suited for development of cryptographic primitives and protocols. Many high-quality libraries exist that can be used off-the-shelf and compiled to WebAssembly.

Concordium also provides additional validation of Rust code so that generated modules conform to on-chain requirements: for instance, that smart contract entrypoints have appropriate types.

Ultimately, however, any language that is able to compile to WebAssembly will be able to target the Concordium chain.

Interoperability

Blockchain interoperability can be considered at different levels such as:

- Inter blockchain ordering of events
- Interfacing to external applications. This could be for retrieving external information to be used in the blockchain ("oracles") or for emitting verifiable information from the blockchain
- Interoperability at smart contract level making it easier to develop decentralized applications based on smart contracts for different blockchain platforms.

Ordering of events across different blockchains can be handled by submitting transactions to the blockchains. However, in practice, such synchronization between blockchains is only as precise as the latency on the different blockchains. The design of the Concordium Consensus layer with almost immediate finalization and thereby protection against roll-back is well suited for such inter blockchain ordering.

Regarding interfacing to other applications, the finalization committee in the Concordium Platform is, in principle, able to create messages authenticated by the blockchain. The format of such outgoing messages is independent of the block structure on the chain and will follow a standardized format. The main hurdle is to create these authenticated messages in such a way that the recipient does not have to travel through the complete chain in order to reach a point of trust in the form of the Genesis Block. Concordium is working on solutions to this.

Another aspect of the external interface is providing external data to the blockchain (decentralized oracles). Concordium implemented one such oracle as it is needed to dynamically update exchange rates in our tokenomics model.

With respect to smart contracts, Concordium has a strategy to use existing programming languages with a strong community of developers in order to lower the threshold for many developers to start working with smart contracts. Furthermore, we expect that the choices of Rust as the initial smart contract languages and Wasm as on-chain languages will make it easier to port smart contracts between different blockchains.

Tokenomics and On-chain Incentivization

The Concordium Platform comprises a set of transactions and economic roles that interact within the economy. An economic role, such as a baker or identity provider, exists either off-chain or is represented by an account on the Concordium Platform.

The flow of CCD between accounts via transactions creates an economy that is designed to incentivize participation in the network and counter dishonest behavior. It is the objective of the Concordium Foundation to guide the creation of a sustainable economy that rewards participants for their efforts in developing the network.

Overview of the Concordium Platform Economy

The Concordium Governance Committee (GC) is responsible for maintaining the health of the economy through monitoring of internal dynamics and scrutiny of the impact of external market conditions. The GC will adopt a flexible approach to nurturing the Concordium Platform economy as it evolves to include more complex dynamics and transactions.

See *Figure 3* below for a visualization of the various roles, accounts and transactions in the Concordium economy, both on-chain and off-chain. In this figure, the blue lines indicate the core flow supported at the moment, while the dotted lines indicate flows planned for future updates.

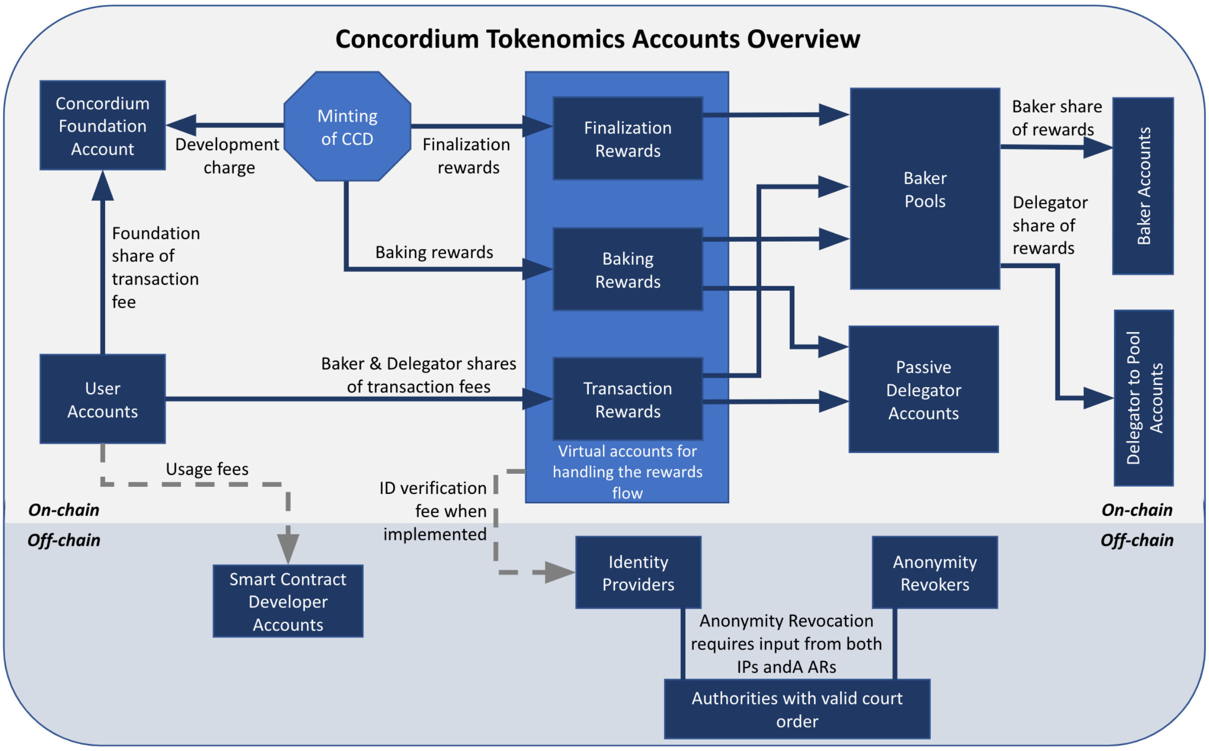
Concordium Tokenomics Accounts Overview
Concordium Foundation Account
Development charge
Minting of CCD
Finalization rewards
Finalization Rewards
Baking rewards
Baking Rewards
Foundation share of transaction fee
User Accounts
Baker & Delegator shares of transaction fees
Transaction Rewards
Virtual accounts for handling the rewards flow
Baker Pools
Baker share of rewards
Baker Accounts
Delegator share of rewards
Passive Delegator Accounts
Delegator to Pool Accounts
Usage fees
ID verification fee when implemented
On-chain
Off-chain
Smart Contract Developer Accounts
Identity Providers
Anonymity Revokers
Anonymity Revocation requires input from both IPs andA ARs
Authorities with valid court order
On-chain
Off-chain

Figure 3: The Concordium Platform economy and roles.

CCD - Concordium’s Native Token

CCD is the native token on Concordium Platform. CCD is a payment token that can be used for a variety of purposes, including as payment for execution of smart contracts, payments between users and payments for commercial transactions. A specific number of CCD is created in the genesis block. After this, the only mechanism to create more CCD is the minting process. The number of CCD that exists on the platform at any time is defined and publicly known.

Roles and Participation in the Economy

There are a number of transactions that allow users and other entities to interact with the economy and each other within the Concordium Platform. Many of these transactions are discussed at length in the section *Execution Layer* above. Specific roles within the Concordium Platform are:

Users can be individuals, businesses, and other identifiable legal entities that create and control accounts within the Concordium Platform.

Users participate in the Concordium Platform economy as follows:

Creating an Account:

1. Account-opening and account-update transactions allow a user to create an account on the Concordium Platform.
2. Initially, there will be no payment required from the user to the identity provider for the creation of a user’s identity object, the costs being covered by the Concordium Foundation to stimulate the ecosystem. However, in the future, users may have to pay a fee to identity

providers, or the identity providers may get a fee as part of the overall tokenomics flow as indicated in Figure 3.

Making transactions:

1. Users, via an associated account, can initiate transactions, including plain transfers of CCD to other accounts, registering information on the chain, or transferring other tokens that exist on the Concordium Platform. In addition to the transactions enabled on the blockchain, smart contracts provide powerful programming functionality to handle specific applications.
2. Users can receive CCD into their account when their address has been specified as the recipient in the transaction.
3. Users are able to make shielded transfers. Anonymous transfers are currently being investigated as another type of transfer that could be added to the blockchain..

Staking:

1. A user can stake part of the CCD on the user's account. This is required if the user wants to operate as a baker.
2. Rewards associated with a user's stake will be automatically transferred to the user's account, and either automatically restaked or made available in the account.

Delegation:

1. In the Concordium Platform, a user can initiate a delegation transaction to delegate an amount of CCD to a baker pool or as passive delegation. Likewise, a user can initiate a similar transaction to un-delegate CCD, i.e., withdraw CCD from the pool or passive delegation.
2. Rewards associated with a user's delegated CCD will be automatically transferred to the user's account, and either automatically re-delegated or made available in the account.

Similar to other users, **the Concordium Foundation** has a number of accounts and runs a number of nodes as bakers. Furthermore, the Concordium Foundation plays a central role in the governance of the chain, including oversight over the Governance Committee that manages the tokenomics parameters as described in the section on Governance. The Concordium Foundation is the recipient of the Platform Development Charge that comprises part of newly minted CCD and part of the transaction fees, which are used for developing the Concordium blockchain and the ecosystem around it.

Bakers are created when users issue special transactions to register as a baker. To register as a baker, a user must stake a certain amount of CCD which is locked (i.e. cannot be spent) while the baker is registered. This stake must exceed a minimum threshold determined by the Governance Committee (currently 14000 CCD). The user also determines whether the baker has a baking pool, allowing other accounts to delegate to the baker.

Finalizers: Bakers automatically become finalizers once they reach a certain threshold of CCD stake. The Governance Committee controls the threshold for how much CCD must be staked (initially set to 0.1% of all minted CCD) and can make adjustments to increase or decrease the number of finalizers with the view of securing the network without compromising speed or efficiency.

Delegators are users who either delegate some CCD to a baker pool or to passive delegation. They receive rewards similar to bakers, but their rewards are lower. Bakers that run pools receive a commission from the delegators to their pool.

Smart Contract Developers: Any user can write, deploy, or use a smart contract on the Concordium Platform. Users that publish a smart contract may also be rewarded for the use of that smart contract by other users. In the future, this process could entail an *App Store*-like library of certified smart contracts. This will allow users to use high-quality code without having to develop it themselves.

Identity Providers perform off-chain identification and create identity objects. Identity providers do not need an account. Identity providers are initially paid in fiat money off-chain but in the future may be incentivized with CCD on-chain for the process of creating identity objects.

Anonymity Revokers perform decryption of encrypted user identifiers related to accounts and disclose such numbers to relevant authorities if presented with a valid court order. Anonymity Revokers do not need an account.

Transaction costs for users

Users of the blockchain pay a Transaction Fee (also referred to as a payment of GAS) for each transaction they make. The fee is paid in CCD, and for that purpose, users will need to acquire and hold CCD.

Transaction costs are designed to be relatively stable in EUR terms, thereby enabling businesses and other users to predict and plan with fixed predictable EUR costs.

With the ENERGY principle described below, Concordium combines the freely fluctuating value of the CCD with a transaction cost that remains stable versus EUR. This is achieved by making the number of CCD to be paid for transactions vary inversely with the price of the CCD when measured against EUR. So if the value of the CCD goes up, a user needs fewer CCD to pay for transactions, thereby maintaining a stable cost in EUR terms. The technical implementation of the above principle is discussed below.

ENERGY, which is an internal measure of transaction cost per transaction, comprises three elements:

Transaction Base Cost (same amount for all transaction)
+ Complexity Premium (dependent on the computational complexity in relation to verification of the transaction and the amount of time it takes to execute)
+ Size Premium for transactions that contain a large amount of data
= Total Transaction Cost expressed as a number of ENERGY

The calculated ENERGY is converted into a CCD cost by applying a fixed EUR/ENERGY conversion rate and a variable CCD/EUR exchange rate which is adjusted dynamically to ensure that the cost remains stable when measured in EUR.

The EUR cost of transactions may be changed by the Governance Committee (see the section on Governance) in response to significant changes in market conditions or changes in the costs related to processing transactions. It is envisaged that over time, the transaction costs will decrease considerably, ultimately approaching zero. Any changes to the EUR cost of transactions will be announced to users and the public with a notice period of no less than 1 month, unless exceptional circumstances occur, thereby enabling users to make necessary adjustments to their use of the blockchain.

The GAS payment in CCD terms is calculated by applying the following calculation:

GAS (CCD) = ENERGY x (EUR/ENERGY conversion) x (CCD/EUR exchange rate)

The CCD/EUR conversion rate required to perform the conversion is supplied using an Oracle that aggregates data from different sources and takes the median of the obtained values. This ensures that if one of the sources provides incorrect data significantly deviating from the other sources, the conversion rate is not affected. Currently, we are using data from CoinMarketCap, CoinGecko, and Live Coin Watch.

Baking and Delegation Rewards

The Concordium Governance Committee is tasked with overseeing and adapting the tokenomics described in this section. They have the possibility of changing the parameters to ensure the long term viability of the economic model.

Sources and Types of Rewards

Bakers and delegators receive rewards for their involvement in the blockchain. There are two sources for these rewards: the transaction fees paid by the users, which are described in the section above, and newly minted CCD.

These sources of CCD fuel three types of rewards:

- **Baking rewards**, which are distributed to a baker pool for every block baked.
- **Finalization rewards**, which are distributed to finalizer pools for being active in the finalization protocol—finalizer pools are also baker pools and receive the baking rewards as well.
- **Transaction rewards**, which are distributed to baker pools for all the transactions included in the blocks that they bake.

Minting of CCD

The process of minting new CCD is an automatic feature of the Concordium Platform whereby newly minted tokens are transferred partly to reward accounts for distribution as incentives and partly to Concordium Foundation. Minting is the only source of growth in the number of CCD in existence inside the Concordium Platform. The growth rate is controlled by the Governance Committee.

Minting is governed by a parameter which controls how many CCD are created per day. This ensures that the CCD growth rate is linked to time and does not depend on fluctuations in the rate by which blocks are created. Initially, the CCD/day parameter is set to have a yearly growth of 10%, and the impact of the CCD growth on the overall tokenomics is monitored to ensure the sustainability of the economy.

A Platform Development Charge of 10% of newly minted CCD is transferred to Concordium Foundation. The remaining 90% of the minted CCD is split between baker rewards (currently 85%) and the finalizer rewards (currently 5%).

Transaction Fees

A platform development charge of 10% is also levied by the Concordium Foundation. The rest of the transaction fees are transferred to the bakers and delegators as transaction rewards. This is described in the next section.

Distribution to Baker Pools and Passive Delegation

Delegation Mechanism

Concordium's delegation mechanism allows a CCD holder to delegate some of their CCD to a baker of their choice. This amount of CCD is added to the baker's pool, which increases the probability of this baker of being selected to bake a block—since this is proportional to the stake in their pool—and thus increases their rewards. Some of the extra rewards obtained by the pool are then distributed to the delegators; the rest goes to the baker.

Concordium has introduced a novel delegation mechanism called *passive delegation*. The delegator who chooses this model does not select a baker to whom to delegate, but instead receives rewards as if they had delegated to all baker pools proportionally to each pool's stake. This has less risk than delegating to a pool since the rewards are not affected by a single baker performing poorly, but the commissions are set so that the expected reward is lower than for delegating to an average baker.

Pay days

Rewards are accrued over a period of 24 hours, and then distributed to accounts. This currently takes place at approximately 8:00 UTC on Mainnet. Changes to baker and delegator stake are also effective at this point (if they were entered at least 1 hour before the end of the pay day).

Baking Rewards

Baking Rewards are distributed to baker pools for every block that they bake. The amount available for baking rewards is currently 85% of the CCD minted in a day, and is divided equally for every block baked. Let this total amount be denoted R_B.

Delegators who choose the passive delegation model get a share which corresponds to the expected reward for delegating to all baker pools proportionally to each pool's stake, minus a passive delegation commission which is currently 12%. This means that an account with passive delegation of s_L CCD will receive

$$R_{B,L} = R_B \cdot s_L / totalStake \cdot 88/100 \,,$$

where $totalStake$ is the sum of the stake of all pools and passive delegators.

Let R'_B denote the baking rewards left after removing the share of passive delegators. And let the number of blocks baked in a day by pool P on the current chain be n_P. Then that pool will receive

$$R_{B,P} = R'_B \cdot n_P / totalBlocks \,,$$

where $totalBlocks$ denotes the total number of blocks added to the current chain in the day.

Finalization Rewards

Finalization rewards are distributed to baker pools that are active in the finalization protocol each day. The amount available for finalization rewards is currently 5% of the CCD minted in a day, and is divided between all active finalizer pools proportionally to their stake. Let this total amount be denoted by R_F.

Finalization rewards are currently not distributed to passive delegators (the commission is 100%). So all the rewards are shared amongst the pools. Let an active finalization pool P have stake s_P and let $totalActiveFinalizerStake$ denote the total stake of all active finalizer pools. Then pool P will receive

$$R_{F,P} = R_F \cdot s_P / totalActiveFinalizerStake.$$

Transaction Rewards

The transaction rewards are distributed to baker pools when they bake a block. As said above, 90% of the transaction fee is available as reward, since the Foundation takes a 10% cut. Let R_T the transaction rewards available from a specific block and let the baker pool P be the one that baked this block.

Like for the baking reward, passive delegators receive a share of the transaction reward corresponding to the expected reward for delegating to all baker pools proportionally to each pool's stake, minus a passive delegation commission which is currently 12%. This means that an account with passive delegation of s_L CCD will receive

$$R_{T,L} = R_T \cdot s_L / totalStake \cdot 88/100\,,$$

where $totalStake$ is the sum of the stake of all pools and passive delegators.

Let R'_T denote what remains after removing the passive delegators' shares. This is, however, not all distributed to the pool that baked the corresponding block: Concordium has introduced a smoothing mechanism, so that blocks with high transaction fees profit the following bakers as well. More specifically, the remaining transaction rewards are shared between the pool that baked the block and a virtual account called the **GAS account**. The current share is 50% for each. Furthermore, the pool that baked the block does not only get 50% of R'_T, they also get 25% of what is currently in the GAS account. We thus have

$$R_{T,P} = R'_T \cdot 1/2 \;+\; GAS_{old} \cdot 1/4\,,$$
$$GAS_{new} = GAS_{old} \cdot 3/4 \;+\; R'_T \cdot 1/2\,.$$

Furthermore, if the block includes account creation and protocol update transactions, the baker pool receives a small percentage of the GAS account for these transactions.

Distribution from Pools to Bakers and Delegators

The section above covered the distribution of rewards to baker pools and to passive delegators. This section explains how the pool rewards are shared between the baker and their delegators. In a first step, every member of a pool receives a share of the pool rewards proportional to their own stake. Then, delegators return a part of the rewards to the baker. This part is called the commission, and can be different for baking, finalizing and transaction rewards. The current values are 10% for the baking and transaction commissions and 100% for the finalization commission.

More specifically, let R_P be the rewards received by a pool in a certain category, and let c be the commission of that category. Let s_P denote the total stake of the pool, let s_B denote the stake of the baker and let s_D denote the stake of a specific delegator D. Then the rewards received by D are

$$R_D = R_P \cdot s_D / s_P \cdot (1 - c),$$

And the rewards received by the baker are

$$R_B = R_P \cdot s_B / s_P + \; R_P \cdot (s_P - s_B)/s_P \cdot c.$$

Relationship between staking, CCD growth, and return on staking

Bakers receive rewards from transaction fees and newly minted CCD. In this section, we explain the relation between the rewards from minted CCD and the amount of CCD staked in the simplified setting with no delegators and without distinguishing between finalizer and non-finalizer bakers.

Because the number of minted CCD per block (and thus the rewards) are linked to the CCD growth while the amount of staked CCD is the result of each individual baker's staking decisions, the Return on Staking (RoS) for bakers and finalizers will fluctuate over time.

As described above, rewards being paid out to bakers and finalizers will be 90% of new minting. If the CCD growth rate is 2%, and 50% of all CCD are staked, then the RoS will be:

2% x 90% / 50% = 3.6%

If the number of staked CCD drops to 25%, the RoS will be:

2% x 90% / 25% = 7.2%

The initial CCD growth rate after mainnet launch is set to 10%. If the CCD growth rate is 10%, and 50% of all CCD are staked, then the RoS will be:

10% x 90% / 50% = 18.0%

If the number of staked CCD drops to 25%, the RoS will be:

10% x 90% / 25% = 36.0%

When the amount of staked CCD goes up, the Return on Staking goes down, and vice versa. This relationship is seen as having a stabilizing effect as it makes it financially more attractive to stake when there is a low amount of CCD being staked and vice versa.

Managing Staking

Baker Status

If a user wants to bake/finalize on a certain day, they must register as a baker at least an hour before the start of that pay day—currently, pay days start around 8:00 UTC on Mainnet.

The baker registration includes the following information:

- An account address. This account is associated with the baker, and only the owner of the account can make changes to the baker. The baker's rewards are paid to this account, and the capital staked by the baker comes from this account.
- Equity capital. This is the amount staked by the baker. This amount must be less than the balance on the account at that point in time. The balance of the account cannot fall below the equity capital while the account is a baker, meaning that the baker's stake is locked and cannot be spent or transferred by the account holder.
- Whether rewards are to be automatically added to the equity capital.
- Whether the baking pool associated with the baker is open to all delegators, closed to new delegators or closed to all delegators.

The size of a baker pool is currently capped at 10% of all stake in pools. If this amount is exceeded (e.g., via automatically re-staking rewards), the excess is not counted towards the lottery power of the pool and does not result in extra rewards. We note that the intention is to decrease this cap in the future.

As long as the baker is active, the equity capital CCD amount is locked in the account and cannot be moved. Any amount above the equity capital can freely be transferred to another account or used for paying transaction fees.

If a baker wishes to change their stake it can be done by updating their baker registration with a higher or lower equity capital. The equity capital CCD amount has to be available on the account for the transaction to succeed, otherwise, the update will fail and the equity capital will not be updated.

If the baker wants to increase their stake for a given pay day, the re-registration must be done at least an hour before that pay day starts.

If the baker wants to decrease the stake, the change will take effect after 3 weeks. During this cool-down period the baker will continue to bake with the original stake (and any restaked rewards). The same applies to stopping baking altogether. During the cool-down period, the baker cannot make further changes to their stake.

A baker can also change whether rewards are automatically restaked, and the status of their baker pool, by updating their baker registration.

Delegator status

A delegator has the choice between delegating to a specific baker or to passive delegation, and can also choose if the rewards are automatically restaked or made available in their account.

Baker pools have two bounds. The first is the bound on the total stake which is 10% of all stake in pools and was mentioned above. The second is that the ratio between the stake in the pool and the stake of the baker running the pool can be at most 3. It is not possible to delegate to a pool if the new stake would violate one of these bounds. But if they are violated by re-staking rewards, then the excess does not count towards the lottery power of the pool and does not produce rewards.

Delegators may also change their stake. Increasing the stake takes effect in the next pay day, if it was entered at least an hour before the end of the pay day. Decreasing stake takes effect after 2 weeks. But changing baker pool or moving to passive delegation is effective on the next pay day.

If a baker shuts down, the delegators are automatically moved to passive delegation.

Governance

The Road in a Nutshell

The purpose of the Concordium Foundation is defined in its Public Deed. The Foundation Board is tasked with ensuring that the Concordium protocol continues to develop and remain relevant to the needs of users within the principles of the Foundation's purpose. The Concordium Foundation is supervised by the Swiss authorities.

After a preparation phase, the Concordium blockchain will be decentralized in two phases by a suitable form of blockchain democracy that will be a hybrid between representative and direct democracy.

In the preparation phase, the Concordium Foundation Board sets up the Governance Committee (GC) that evaluates and implements parameter changes and protocol updates, and oversees and adjusts the tokenomics.

In the first phase of decentralization, CCD holders will assume an increasing role in suggesting and determining the priorities for the Concordium blockchain development through the Governance Committee. They will be able to elect members to the GC and propose candidates for the GC. In addition, they will obtain the right to vote on GC proposals and will have the opportunity to formulate their own proposals. The Foundation Board will act as a guarantor that the fundamental Concordium principles of privacy with accountability and the Public Deed of the Concordium Foundation are adhered to.

In the second phase of decentralization, by allowing collective decisions of CCD holders to change the governance framework from the first phase, CCD holders may assume an even stronger role in shaping and overseeing the development of the Concordium blockchain.

Preparation phase: launch of Mainnet in June 2021 to June 2024.

In this preparation phase, the Concordium Foundation Board appoints a Governance Committee with five members, i.e., the GC is 100% appointed by the Concordium Foundation Board.

The Governance Committee has two tasks. First, it evaluates and implements parameter changes and protocol updates, and oversees and adjusts the tokenomics, including the CCD growth rate within the discretion granted by the Concordium Foundation. Second, the GC develops the details of the governance framework for the start of the first phase and submits its proposal to the Foundation Board for approval. Besides legal and organizational aspects on how the GC itself should operate, the GC ensures that conceptual and technical solutions are available to elect members to the GC.

In this preparation phase, the Foundation Board oversees the work of the GC.

First phase of decentralization: June 2024-June 2027

June 2024 (Mainnet +3 years)

CCD holders appoint two members of the Governance Committee by electronic voting, expanding the Committee from five to seven members. At this stage, 28.6% of the GC is appointed by the CCD holders.

The Concordium Foundation will update its guidelines for the GC and the GC continues with its first task, i.e., the GC is responsible for parameter changes, protocol updates and the tokenomics. The Foundation board continues to supervise that the GC respects the guidelines and can intervene if not.

In addition, the GC prepares a framework and a technical solution in which CCD holders themselves can propose candidates to the GC. Moreover, the GC prepares a framework and a technical solution for how CCD holders can vote on proposals of the GC.

June 2025 (Mainnet +4 years)

CCD holders appoint two further members of the Governance Committee by electronic voting, expanding the GC from seven to nine members. All candidates for the GC will be proposed by CCD holders. At this stage, 44.4% of the GC is appointed by the CCD holders.

In addition to the right to vote for GC members introduced in 2024, token holders can now also vote on proposals from the GC. The governance framework will determine what proposals are put to a vote and what quorums are needed for them to be accepted.

In addition to the standard task (parameter changes, protocol updates and the tokenomics), the GC will prepare a framework and a technical solution for how CCD holders can make proposals themselves to change parameters or elements of the tokenomics.

June 2026 (Mainnet +5 years)

From this point on, the Committee's total membership remains at nine members, with three seats up for (re)election every year. Thus, three of the initial five committee seats appointed by the Foundation Board get confirmed or replaced by CCD holders in 2026, and 77.8% of the GC is appointed by the CCD holders at this point.

In addition to the right to elect GC members and vote on GC proposals, CCD token holders obtain the right to make their own proposals to change parameters or elements of the tokenomics.

Second phase of decentralization: from June 2027 onward

June 2027 (Mainnet + 6 years)

After these elections, 100% of the Governance Committee will be appointed by the CCD holders.

As part of the second phase of decentralization, certain aspects of the governance framework may be modified by token holders by allowing collective decisions of CCD holders to change the governance framework from the first phase. CCD holders may assume an even stronger role in shaping and overseeing the development of Concordium. How far the shift of power from the Foundation Board and the GC to the CCD holders on fundamental governance decisions will be and how much CCD holders also assume an oversight role on the GC will be determined in 2027, depending on the experience on the first phase and on the scientific research on this issues at that time.

References

An up-to-date list of research papers from the Concordium Blockchain Research Center Aarhus (COBRA) can be found under https://cs.au.dk/research/centers/concordium/publications/.

[ANS19] Annenkov, D., Nielsen, J., Spitters, B. "ConCert: A smart contract certification framework in Coq." Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, 2019. https://doi.org/10.1145/3372885.3373829.

[CM19] Chen, J., Micali, S. Algorand: A secure and efficient distributed ledger, Theoretical Computer Science, Volume 777, 2019. https://doi.org/10.1016/j.tcs.2019.02.001.

[DGKOS21] Damgård I., Ganesh C., Khoshakhlagh H., Orlandi C., Siniscalchi L. (2021) Balancing Privacy and Accountability in Blockchain Identity Management. Topics in Cryptology – CT-RSA 2021. Lecture Notes in Computer Science, vol 12704. Springer, Cham. https://doi.org/10.1007/978-3-030-75539-3_23

[DGKR18] David B., Gaži P., Kiayias A., Russell A. Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. Advances in Cryptology – EUROCRYPT 2018. Lecture Notes in Computer Science, vol 10821. Springer, Cham, 2018. https://doi.org/10.1007/978-3-319-78375-8_3.

[DMMNT20] Dinsdale-Young T., Magri B., Matt C., Nielsen J.B., Tschudi D. "Afgjort: A Partially Synchronous Finality Layer for Blockchains." Security and Cryptography for Networks. SCN 2020. Lecture Notes in Computer Science, vol 12238. Springer, Cham. https://doi.org/10.1007/978-3-030-57990-6_2.

[DMMNT22] David, B., Magri, B., Matt, C., Nielsen, J.B., Tschudi, D. "GearBox: Optimal-size Shard Committees by Leveraging the Safety-Liveness Dichotomy." 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22). Association for Computing Machinery, New York, USA. https://doi.org/10.1145/3548606.3559375.

[DSTT19] Dinsdale-Young, T., Spitters, B., Thomsen, S., Tschudi, D.: WIP: Formalizing the Concordium consensus protocol in Coq. CoqPL 2019. https://cs.au.dk/~sethomsen/coqpl19.pdf.

[GOT19] Ganesh C., Orlandi C., Tschudi D. (2019) Proof-of-stake protocols for privacy-aware blockchains. Advances in Cryptology – EUROCRYPT 2019. Lecture Notes in Computer Science, vol 11476. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-17653-2_23.

[Kwo14] Kwon, J. Tendermint: Consensus without mining. Manuscript, 2014. https://tendermint.com/static/docs/tendermint.pdf.

[KMM+21] Kamp S.H., Magri B., Matt C., Nielsen J.B., Thomsen S.E., Tschudi D. (2021) Weight-Based Nakamoto-Style Blockchains. Progress in Cryptology – LATINCRYPT 2021. Lecture Notes in Computer Science, vol 12912. Springer, Cham. https://doi.org/10.1007/978-3-030-88238-9_15

[NS19] Nielsen, J., Spitters, B.: Smart contract interactions in Coq. CoRR abs/1911.04732, 2019. https://arxiv.org/abs/1911.04732.

[Per16] Perrin, T. The noise protocol framework, 2016. http://noiseprotocol.org/noise.pdf.

[TS21] Thompson S. E., Spitters B.: Formalizing Nakamoto-Style Proof of Stake, 2021 IEEE 34th Computer Security Foundations Symposium (CSF), doi: https://doi.org/10.1109/CSF51468.2021.00042.

DISCLAIMER. This document reflects the Concordium Foundation's current plans regarding the functionality and specifications of the Concordium Platform, technology, blockchain and ecosystem (the 'platform'). Readers should be aware that: a) the actual and further development of the platform may deviate from what is described in this document; b) that the Concordium Foundation may change the specifications of the platform at any time; and c) that the Concordium Foundation accepts no liability for the reliance by third parties on the information in this document.