Q: What is Crust?

Crust implements the incentive layer protocol for decentralized storage with adapting to multiple storage layer protocols including IPFS, and provides support for the application layer. At the same time, Crust's architecture also can provide support for a decentralized computing layer to build a distributed cloud ecosystem.
Decentralized storage allows files to be stored on different hosts by distributing the files across the entire network without being restricted by any centralized entity. Users can flexibly manage their data and effectively enhance data security and reliability through data encryption, secure backup, authorized access and other means. At the same time, the storage space sharing model can also effectively reduce storage costs. In 2019, Gartner, the world's leading research and advisory company, announced the Top 10 strategic technology trends for 2020, which shows that distributed cloud service and practical blockchain technology are among the top ten technology development directions in 2020. Crust is exactly the overlap and extension of these two areas.

Q: In which scenarios can Crust be applied?

Crust's decentralized storage layer provides a distributed file system. At the same time, Crust encapsulates some standard interfaces such as Amazon S3-like. Any application scenarios involving data storage, such as cloud services, edge computing, and decentralized applications, are the scenarios that Crust can adapt. Worth mentioning is that in edge computing scenarios, compared to centralized cloud storage, Crust's decentralized storage is closer to the edge, which can achieve relatively low cost and high performance.

Q: What are the main differences between Crust and Filecoin, a star project in the storage field?

First is about the difference in complexity and cost. To be more detailed, the incentive for storage providers is a key issue in the field of decentralized storage. A fair incentive mechanism must be that the more storage provided and the greater the contribution, the more rewards can be obtained. So the question turns into how to make the blockchain accurately reach a consensus on the volume of storage provided by each node, and by so to reward those who perform storage normally and punish those who perform storage improperly, such as those who delete user files without permission. Further, the key issue of storage incentives has evolved: how storage nodes prove their storage capabilities. An important difference between Filecoin and Crust lies in the way of proof of storage. The storage capacity of Filecoin requires each node to periodically provide proof to the network. This proof is calculated by the node through a zero-knowledge proof based on the file it stores. However, since the Crust node supports the Trusted Execution Environment (TEE) technology, it can check the file locally on the node and periodically generates a workload report signed by the TEE. Crust's local inspection method does not need to introduce the complex zero-knowledge proof, which reduces computational costs and lowers thresholds. At the same time, compared to the periodic report generated by zero-knowledge proof, Crust's workload report is much smaller, which decreases the occupation of network resources.
Second is the difference between incentive mechanisms. Filecoin's current incentive mechanism for storage nodes is based on the number of storage orders accepted by the node, while Crust's incentive is based on the storage space or capability provided and the collateral obtained by the node, which is what we call staking. Crust's incentive for storage nodes is independent of the storage market, which is conducive to the growth of storage resources during the cold start period.
Then there is the difference in the consensus of producing blocks. Filecoin is using an EC (Expected Consensus) algorithm based on VRF (Verifiable Random Function) and VDF (Verifiable Delay Function), while Crust is based on GPoS, a PoS consensus where storage resources are used as guaranty quota.
Last but not least, the difference between Crust and Filecoin is also reflected in the support for computing. Crust's trusted resource monitoring module, in addition to monitoring and quantifying storage resources, also can monitor and quantify computing resources. The second-tier computing consensus module and integration for cloud computing applications are also included in Crust's future plans.

Q: How does Crust network quantify storage workloads provided by nodes?

Based on the TEE (Trusted Execution Environment) technology, Crust implements MPoW (Meaningful Proof of Work) mechanism to quantify meaningful storage resource usage and generate corresponding work report in a reliable way.
TEE is the abbreviation of the Trusted Execution Environment. It is a secure area on the main processor, which can guarantee the security, confidentiality, and integrity of the code and data loaded into the internal environment. TEE provides an isolated execution environment. The security features provided include isolated execution, the integrity of trusted applications, the confidentiality of trusted data, and secure storage.
In Crust, when each node enters the network, the entire network needs to verify the identity of its TEE. It is determined that it is a valid TEE, and then remote authentication is performed to determine that the validation logic is executed in the TEE. Then this TEE is equivalent to the node's monitor, which periodically checks the working

status of the check node and generates a working report. Other nodes can reach a consensus on the storage space by verifying the TEE signature of the work report.

Q: What are the consensus mechanism and economic model design of Crust?

Crust is using the GPoS (Guaranteed Proof of Stake) consensus, which is called PoS consensus with storage resources as a guaranty quota. The storage resources mainly refer to meaningful data stored and empty disks available.Similar to existing PoS projects, nodes need to stake CRU tokens to compete for the right to generate blocks. The difference is that nodes also need to provide storage resources to obtain the corresponding guaranty quota. With the guaranty quota, the corresponding number of CRUs can be staked.
Under this mechanism, two types of assets, storage resources, and CRU tokens are must conditions to become nodes. By combining the advantages of resource-based (such as Bitcoin) and token-based (such as Cosmos) consensus mechanisms, network security can be more effectively protected from malicious deeds. If Crust network consensus is under attack, in addition to requiring a large percentage of CRU tokens, you also need to be able to control a sufficient amount of storage resources. This design makes the attack very difficult.
The node also allows the guarantor to use the CRU as a guarantee on the premise that it owns the storage resources as a guarantee. That is, the node-staked CRU can be it's own or from a guarantor. When the guarantor chooses to guarantee the node, it allows guarantors to obtain benefits, but also requires them to bear the risk of the node being punished. If a node is confiscated by the system for triggering a penalty mechanism, the guarantor will also be fined at a certain rate since it serves as a guarantor. Under this mechanism, the guarantor will tend to choose a node with good faith and good service to guarantee it, and the market will decide a balance between the guaranteed income and the risk of punishment
As a network protocol for underlying data storage, Crust provides the function of storing resource transactions. CRU tokens are used as the contract security deposit in this trading market to ensure the order of the trading market.
Finally, similar to other blockchain projects, CRU tokens will also be served as transaction fees for using the network, which is much similar to Gas in Ethereum. It can also be used directly to purchase resource services in the network.

Q: Why did you choose Substrate?

First of all, the technical framework of the Substrate is excellent, and it is very friendly to the performance and functional support of Crust, the application-type blockchain. Offchain Worker also well supports the implementation of the Crust storage market.
Second, because Crust's storage workload consensus is based on TEE technology, and TEE technology has multiple solutions, such as IntelSGX, AMD SEV, and ARM Trustzone. Crust's autonomous mechanism based on Substrate can be used to maintain the TEE solutions list.
Finally, the XCMP protocol brings possibilities to Crust's cross-chain ecosystem. Imagine that the storage service provided by Crust in the future can be purchased with multiple tokens, and any project in the ecosystem can use the storage capabilities brought by Crust. This is a combination of 1 + 1 and results greater than 2.

Q: How does Crust join the Web3 / Polkadot ecosystem, and what impact will it have?

From the perspectives of function and technology, Crust's positioning in the Web3 protocol stack belongs to "Data Distribution Protocols". The Crust’s storage layer protocol is capable of providing encrypted storage and support for large calculations required by upper-layer protocols.
Crust's upper layer design includes the storage market, retrieval market, and file management mechanisms. Naturally, it needs to be compatible with some decentralized data transmission protocols, applications, and financial systems, and these protocols and applications have been fully implemented in the Web3 ecosystem. For example, some file protocols in the Web3 ecosystem, such as DAT, Shift, etc., can provide flexible file management functions and better support version controls of dynamic files, but they all require a storage incentive layer such as Crust to provide the most basic storage resources.
Finally, the Crust project will join the Polkadot ecosystem as a parallel chain, effectively supporting projects and applications within the ecosystem. Because Crust is technically closely integrated with projects such as IPFS and Phala, Crust can also effectively expand the technology ecosystem of Web3.

Q: Why did Crust choose TEE technology?

The regular inspection of TEE is equivalent to the monitoring tool provided by each storage node. Of course, there are many ways to implement this monitoring function. TEE technology is not a proprietary hardware technology. Intel SGX and ARM TrustZone have released TEE technology in the latest CPU series. Various Intel x86 series and ARM computers are equipped with TEE. Software TEE (TEE implementation with TPM chip) is also being developed. Therefore, Crust chose TEE, which is an optimal solution obtained after balancing complexity, stability, and cost.

Q: What are the recent developments and plans for 2020?

In March 2020, we launched Crust Alphanet, and at the same time we will open some source code to the public. Crust has recently passed the review of the Substrate Builder's Program and will receive technical support and ecological assistance from the Substrate team.
We plan to launch Crust Betanet in June of the same year. In this version, Crust will support the full-featured economic model and storage trading market, and we plan to join the Phala testnet to provide decentralized storage for Web3 applications.
The Crust mainnet is planned to go online and access Polkadot slots in October 2020.

Q: File sharing is a common cloud storage scenario. Will file sharing links be available on Crust?

Yes. The next version of Crust Cloud will include file-sharing links and code-extraction capabilities.

Q: Will Crust Cloud support the storage of private files in the future?

Yes, the next version of Crust Cloud will offer a privacy storage function, where files will be uploaded encrypted and inaccessible. Further, we plan to upgrade the Crust Network before and after the main network launch, so that nodes can support TEE based data authentication and encryption and thus provide full flow protection for user privacy.

Q: During the process of Crust Cloud storage, are files redundant and backed up? Are files stored in shards?

Backup orders are already supported in the Crust storage market. Automatic node selection and initiation of multiple replicas will be implemented in future Crust Cloud versions.
Sharding: Generally distributed file systems such as FastDFS and IPFS Cluser can both support sharding within a cluster. Crust is currently compatible with both the FastDFS and IPFS file protocols and can efficiently support both single point and cluster morphology of nodes. Further, Crust Network currently supports shard storage orders, and in the future Crust Cloud will accordingly support shard storage for large files across nodes by initiating shard storage orders.

Q: Is IPFS used on Crust? Is multi-point service supported?

IPFS is currently available for miners' internal storage, but IPFS routing has not yet been enabled in Crust Network. The main reason is that IPFS is not yet stable enough relative to the large Internet ecosystem. Crust will be closely monitoring the develop progress of IPFS. As for multi-point download services, some P2P protocols are well supported except for the IPFS stack. We will closely monitor and evaluate these protocols as well and make compatibility upgrades in the future. Currently, the Crust team is focused on basic storage proof.