REGISTRATION SYSTEM BY THE BLOCKCHAIN

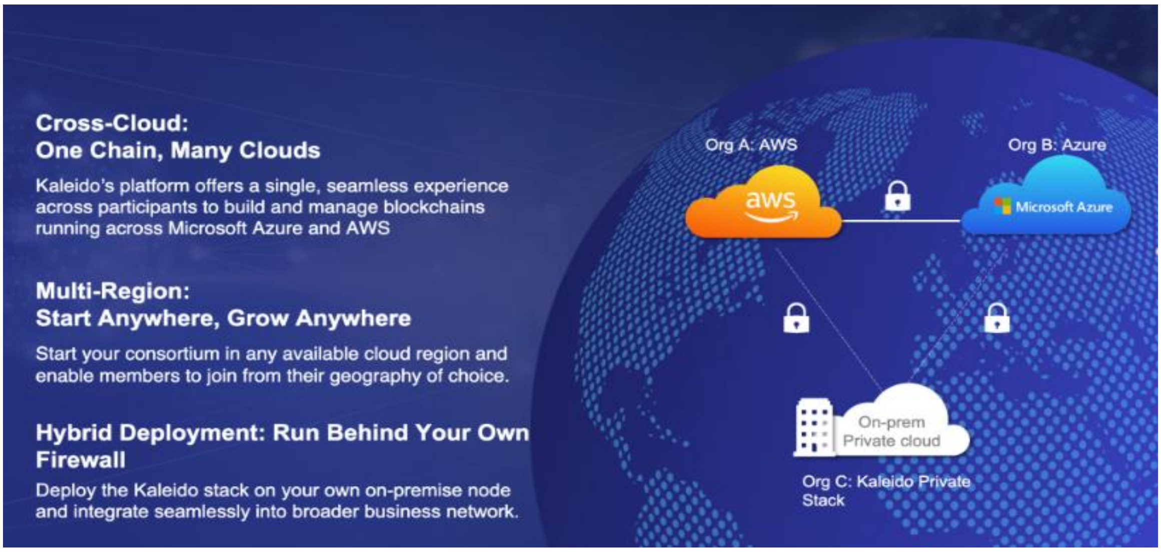
Blockchain System via Kaleido (US provider) is a full stack SaaS platform that’s hybrid enabled and designed to simplify the process of building consortia and deploying private blockchain networks. The service provides a “permissioned” implementation of the Ethereum protocol, whereby member participants operate with authenticated identities backed by digital certificate chains. The trust and transparency delivered by the Kaleido platform allows for the finality and speed of transactions to be maximized through the use of efficient consensus algorithms otherwise unavailable in public/anonymous chains. Environments can be configured to run against one of three consensus algorithms – Proof of Authority, Raft or Istanbul BFT – and both public and private transactions are supportable by means of the Quorum client and its corresponding Tessera module. This protocol and transaction class flexibility is especially powerful for enterprise orchestrations, where oftentimes not all participants are entirely trusted and certain pieces of data must be obscured from the overall network.

Users have the option of deploying blockchain resources in AWS, Azure or on-premise, thereby maintaining business continuity with their hardened business processes and existing IT estates. Native cloud resources, such as key management services and log streaming can be woven into the blockchain runtime for heightened control and extended functionality.

The Kaleido Marketplace exposes a library of powerful ancillary chain layer services, B2B connectors and 3rd party solution accelerators, and offers unified access to the critical building blocks that encompass an enterprise production-caliber solution. Examples of marketplace services include integration gateways to legacy systems, app to app messaging pipes, off-chain file storage, customizable oracles and more.

Borderless Blockchain

Diverse business networks bring their own pre-existing investments in cloud, IT structures, operating preferences and data residency requirements. Decentralized ownership of the network across clouds, regions, and companies’ private nodes is a foundational enterprise blockchain requirement. The Kaleido platform offers hybrid deployment to uniquely fill this need, providing a truly integrated and seamless experience across Amazon Web Services, Microsoft Azure, and on-premise data centers that enables borderless blockchain for an accelerated pace of adoption by global networks. Start on your cloud of choice and expand your network seamlessly and securely with a single experience across leading cloud providers and geographic regions, all on Kaleido.



Cross-Cloud:
One Chain, Many Clouds

Kaleido’s platform offers a single, seamless experience across participants to build and manage blockchains running across Microsoft Azure and AWS

Multi-Region:
Start Anywhere, Grow Anywhere

Start your consortium in any available cloud region and enable members to join from their geography of choice.

Hybrid Deployment: Run Behind Your Own Firewall

Deploy the Kaleido stack on your own on-premise node and integrate seamlessly into broader business network.

AWS

The following hosting regions are available on AWS cloud infrastructure. Each region contains three underlying availability zones for fully-managed high availability, disaster recovery, synchronized file system replication and preemptive auto-scaling:

- US-East-2 (Ohio)
- EU-Central-1 (Frankfurt)
- AP-Southeast-2 (Sydney)
- AP-Northeast-2 (Seoul)

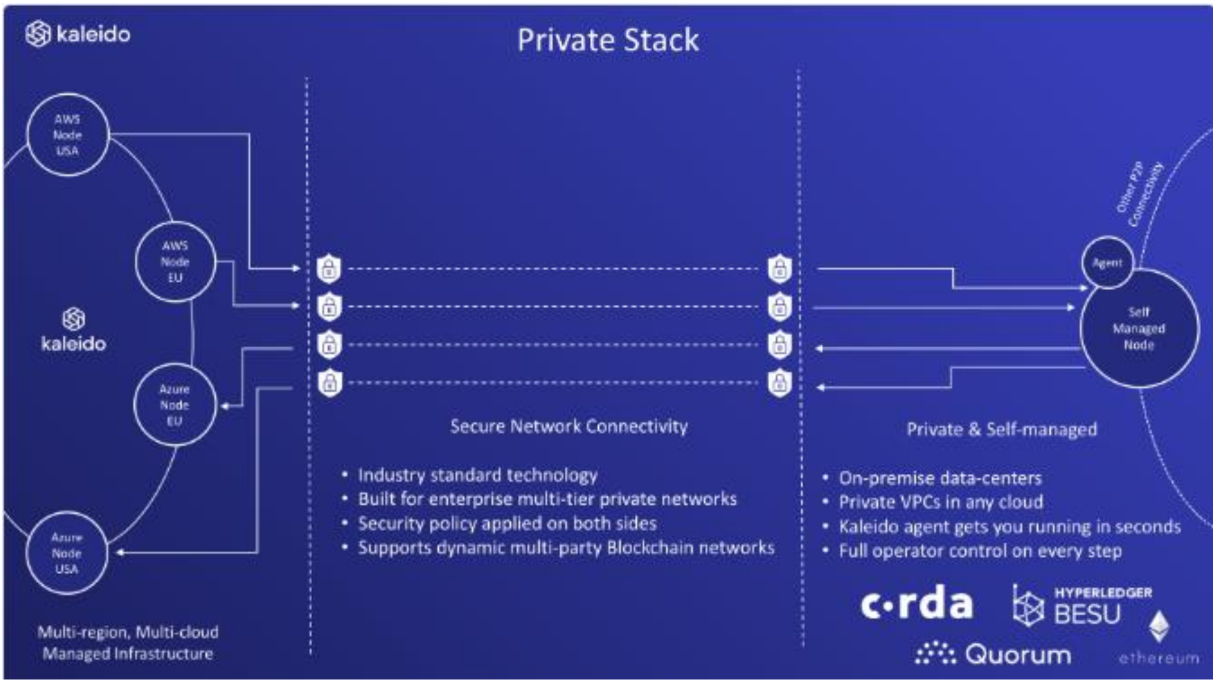
Microsoft Azure

The following hosting regions are available on Microsoft Azure cloud infrastructure. Each region contains three underlying availability zones for fully-managed high availability, disaster recovery, synchronized file system replication and preemptive auto-scaling:

- West-US-2 (Washington)
- **NEW** France-Central (Paris)

Hybrid deployment with PrivateStack

Seamlessly connect nodes running self-managed on-premise, or in private cloud VPCs, with nodes running fully-managed across AWS and Azure in Kaleido. Our unique blockchain-aware PrivateStack network bridge and remote agent combine industry standard network technology, with DMZ-ready dynamically configured networking, and one-click setup.



Private Stack

Secure Network Connectivity

- Industry standard technology
- Built for enterprise multi-tier private networks
- Security policy applied on both sides
- Supports dynamic multi-party Blockchain networks

Private & Self-managed

- On-premise data-centers
- Private VPCs in any cloud
- Kaleido agent gets you running in seconds
- Full operator control on every step

Multi-region, Multi-cloud Managed Infrastructure

The PrivateStack software package provides you with two key components:

PrivateStack Network Bridge

Combines forward & reverse proxies, secure network bridging via industry standard DMZ friendly networking, and dynamic configuration management. Designed to put you in full control of your network security, while retaining bi-directional connectivity to and from the whole business network.

PrivateStack Agent

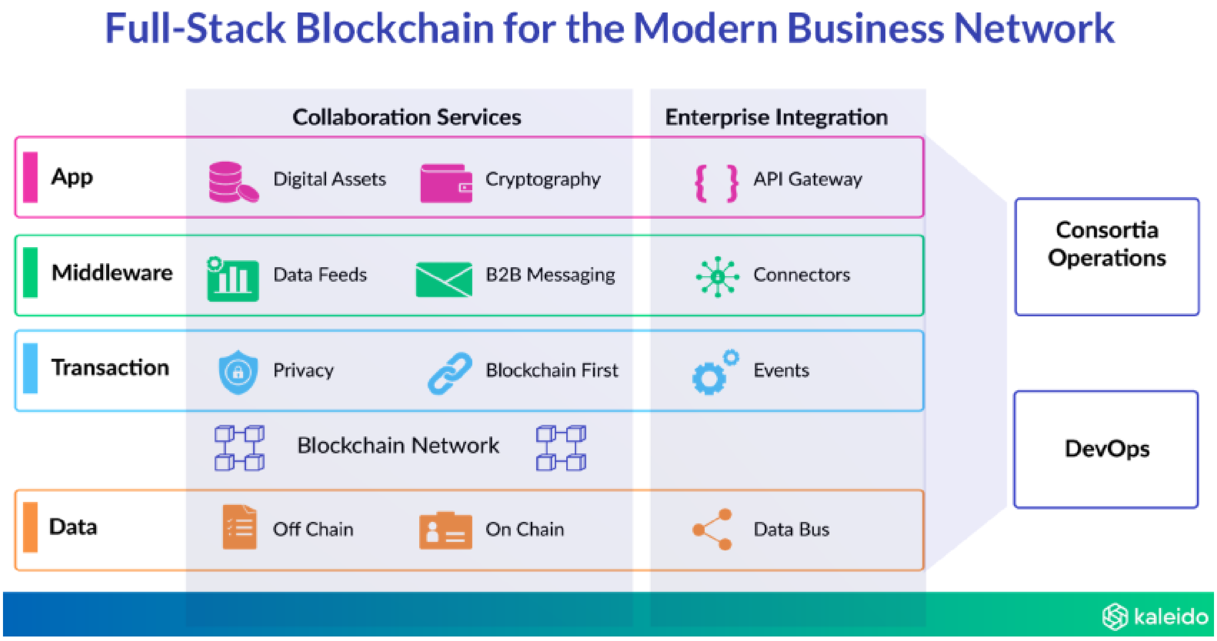
Simplifies the creation of your private self-managed node into a single command. Provides a secure API to bootstrap your node into the business network, and share its public identity with other nodes in the same Kaleido environment. A small cut-down adapted from the version of the agent we run for nodes in the Kaleido fully-managed platform, it provides your organization with enough connectivity to ease on-boarding, without compromising the integrity of your private keys and data.

What Is Full-Stack Blockchain?

Enterprise Decentralized applications (DApps) have a fundamentally different architecture and design to traditional applications. With the shared ledger of Blockchain at their core, they communicate on-chain and off-chain between multiple participants to orchestrate end-to-end business transactions, and capture some data as immutable proofs.

They build on core Blockchain programming constructs like 'tokens' to link assets and data from external systems to a unique 'hashes' on chain. Actions and data are pinned to organizational identities, sometimes masked for privacy, which can be traced back to real business entities permissioned into the Business network.

The Kaleido B2B application stack is built from experience of taking modern business networks into production, giving the full suite of tools Enterprises need to realize the potential of Blockchain for their usecase.



Identity

Identity is the foundation of a private and permissioned Blockchain network. It encompasses the following:

- Who has permission to replicate the ledger
- Who has permission to propose and sign blocks
- Who has permission to sign and submit transactions
- Who has permission to interact with smart contracts
- Who has permission to connect into the system
- Who has the ability to extend invitations

Kaleido builds in technical controls at each level allowing you to govern the participants of the business network, and pass on permissions from existing members of the network to new members as they join.

This applies not just to the on-chain data and transaction, but also to the stack of services that sit alongside the chain to make up the B2B stack.

However, none of this is relevant without confidence of the real business entity that is behind a public key based identity being used at runtime.

This is where an infrastructure is critical for binding verifiable organizational identity to keys and other technical identities used in the system.

In the Enterprise space, this usually means leveraging established PKI systems of trust that can be used to easily assert organizational identities, and then binding those identities to the root of a Blockchain backed registry controlled by the participants.

The process for establishing this membership identity requires work from each participant to generate a signed X509 certificate with an appropriate PKI trust chain to be trusted by other participants in the network.

As always, Kaleido provides an Easy Button to let you develop and experiment in early project phases, as well as the solution required for Enterprise production use. With the click of a button you can use use identities self-signed on Kaleido *without a PKI trust chain attached*.

Transaction Submission

The transaction pooling/execution logic within an Ethereum node is based upon the concept of a nonce, which must be incremented exactly once each time a transaction is submitted from the same Ethereum address. There can be no gaps in the nonce values, or messages build up in the queued transaction pool waiting for the gap to be filled (which is the responsibility of the sender to fill). This allows for deterministic ordering of transactions sent by the same sender.

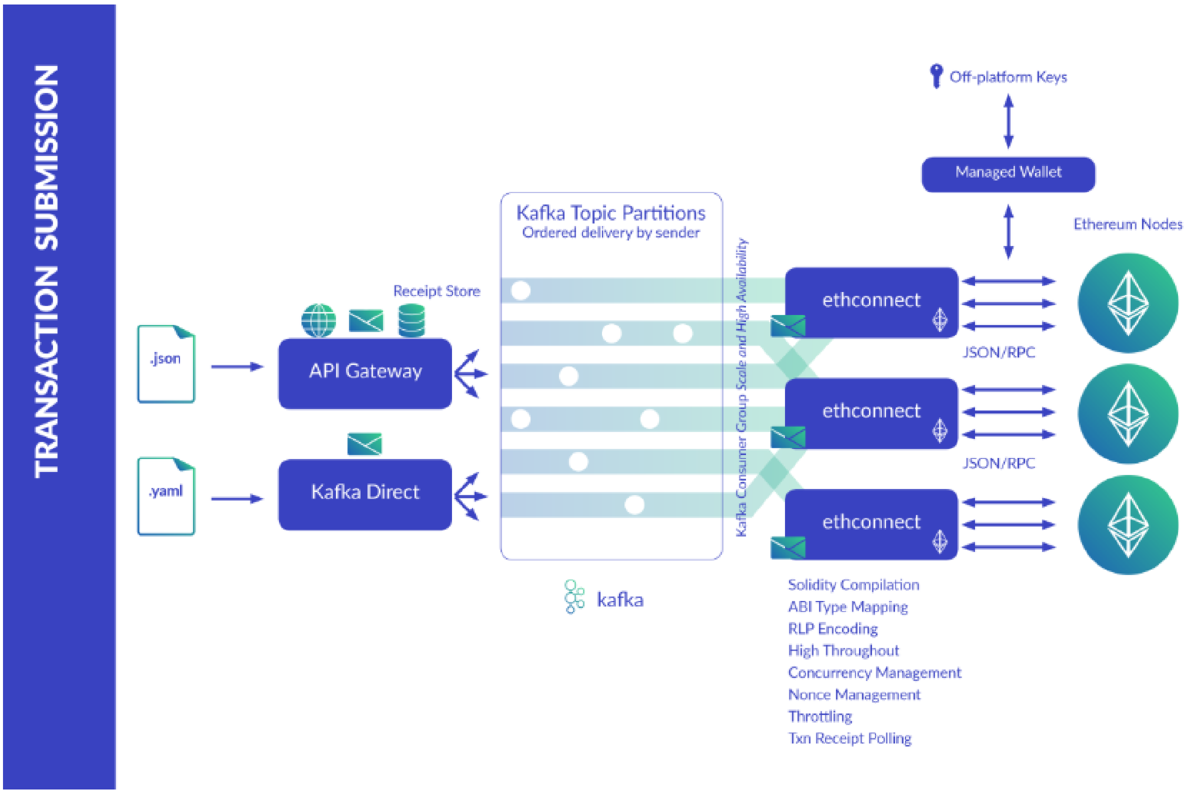
The management of this nonce pushes complexity back to the application tier - especially for horizontally scaled Enterprise applications sending many transactions using the same sender, potentially workload balancing that traffic across multiple Blockchain nodes.

Transaction throughput is also commonly uneven, with periods when the rate of submission of transactions exceeds the efficient transaction-per-block rate of the chain. Flooding these transactions into the Blockchain nodes often results in discarded transactions, because the transaction pools of the individual nodes reach capacity. Or it can result in packing very large numbers of transactions into blocks, resulting in long block times and low efficiency of the network - with reduced overall throughput.

Maybe, counterintuitively, the way to ensure consistent high performance of the chain is to smooth out these peaks in workload, using the tried and tested Enterprise approach of message queuing.

As the above simplified summary shows, the complexities of transaction submission and nonce management can be a significant burden on projects, and attempting to build a bespoke solution is an inefficient task.

Kaleido provides a built-in fully managed Kafka streaming tier, combined with our EthConnect transaction submission and throttling technology - the core of which is contributed to the community as open source.



The EthConnect and Kafka transaction streaming solution is fully integrated with our REST API Gateway, so that applications can benefit from Enterprise Grade production throughput and resilience, with the consumable REST APIs that allow you to get started with the Kaleido platform in minutes.

Off-chain compute

A straight forward way to maintain privacy, is to execute all logic and state transitions off-chain in environments where there is safe access to the data. After-all that's how transactional systems have been working for decades. The questions in the context of a decentralized solution are:

- How to invoke that off-chain processing of the data as part of the transaction
- The level of agreement on the outcome that is required in order for the transaction to complete

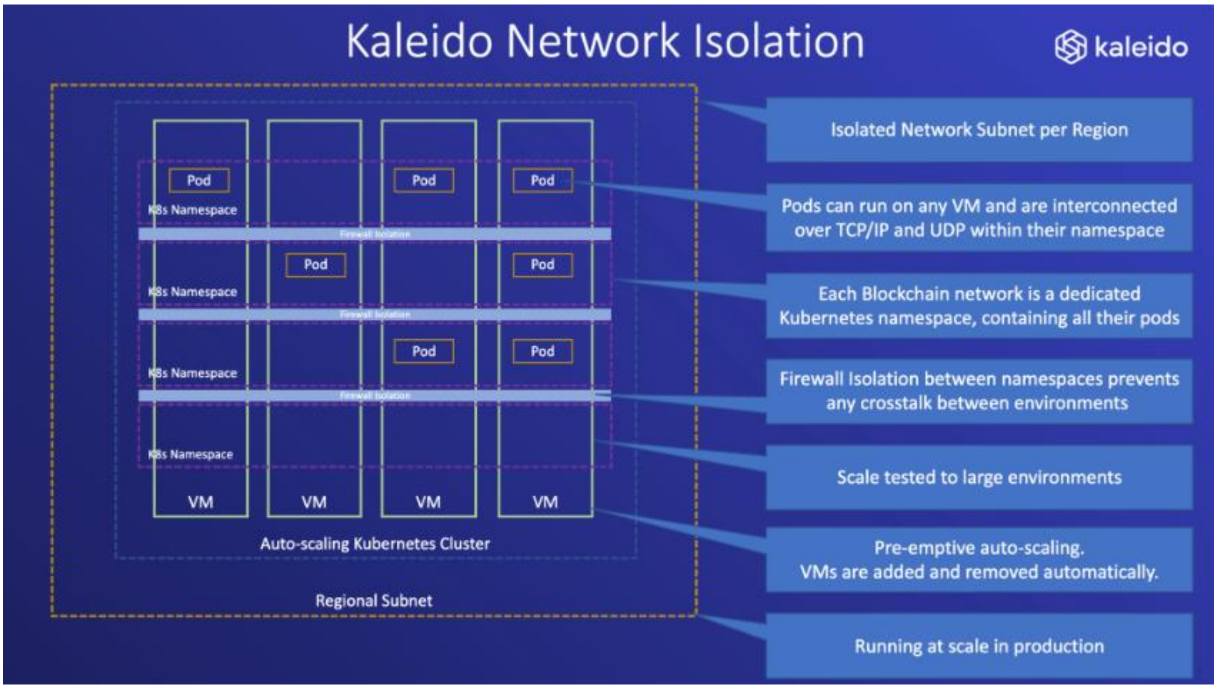
The answers to these questions are always going to be dependent on the business network. In a large number of cases, you can fully meet the requirements of the network using trusted bilateral communications (such as with App 2 App Messaging) coordinated with on-chain transactions at specific agreed and verifiable points. In fact this is extremely common in cases where the private processing is specific to the core systems of each business.

Take a simple example of a private off-chain action - *Verify the authenticity of this Passport as part of a know your customer (KYC) use case*. This might need to trigger a different business process for each participant, with different amounts of automation and human verification in each enterprise. All individually conforming to their audited processes and regular requirements, but unique to the structure and systems of each business. The data for the passport itself is almost certainly not on-chain. Instead maybe it's shared via a private documents store, or distributed as an encrypted payload over an IPFS decentralized file system.

It is possible to go a step further, and coordinate these private off-chain activities as mandatory steps initiated by on-chain Smart Contract logic. We talked about Chainlink in the Oracles section, as a robust system for triggering external processing off-chain in response to on-chain Smart Contract logic, and then bringing some data back onto the chain. In these more sophisticated cases, you need to consider the level of verification you make of the off-chain processing.

A simple example that works for many private permissioned systems, is that the participant performing the action provides a signed proof that they performed it. This could be as simple as a transaction signed by their address, stating their compliance, or the answer to a simple question where the answer does not leak sensitive information (yes/no or threshold based). Here there is trust and/or accountability for answering correctly, tied back to the known organizational identity responsible in the business network.

Platform Scale



Kaleido Network Isolation

Isolated Network Subnet per Region

Pods can run on any VM and are interconnected over TCP/IP and UDP within their namespace

Each Blockchain network is a dedicated Kubernetes namespace, containing all their pods

Firewall Isolation between namespaces prevents any crosstalk between environments

Scale tested to large environments

Pre-emptive auto-scaling.
VMs are added and removed automatically.

Running at scale in production

There are some more elaborate schemes maturing in the blockchain space to provide enhanced proof that off-chain processing was executed correctly.

Trusted Execution Environments (TEEs)

A TEE provides evidence, an *attestation*, that proves the exact code that is being executed. The TEE is also able to lock sensitive data inside of itself, such as an encryption key generated inside of the TEE, so that it can perform processing on sensitive data without it being available, even in memory, to other processes on the machine. A TEE requires special hardware, setup and code to be available for the execution, so there are some setup complexities required to incorporate TEE-based execution in your solution. The emerging value here is the ability for shared compute to execute rich and complex off-chain processing with high throughput, and verified outcomes. This means a single execution off-chain of attested code in a TEE can give equivalent assurance of the outcome of a transaction to executing it many times by the nodes of the chain in on-chain Smart Contract logic.

Multi Party Compute (MPC)

In MPC processing the computation is distributed between multiple parties, where no individual party has access to the full data. This can allow summary calculations to be performed across a large data set, without revealing sensitive data about the individual pieces of data in that data-set. This approach has seen most adoption in cases where individuals have self-sovereign control over their own data, and wish to provide limited access to that data for research, analytics, etc.

Zero Knowledge Proofs (ZKP)

You might have heard of zero-knowledge proofs such as zk-SNARKs (which powered Zcash), zk-STARKs, Range proofs, Bulletproofs and Sigma-Bullets. These are sophisticated mathematical techniques that can be used to generate mathematical proofs that computation has been performed correctly, without divulging the details or data of that computation. A simple example would be asking a question such as "Is the number between 10 and 20?", where off-chain compute that has access to a private number could generate a verifiable proof that the number is (or isn't) in that range, and submit that proof back without revealing the actual value of the number.

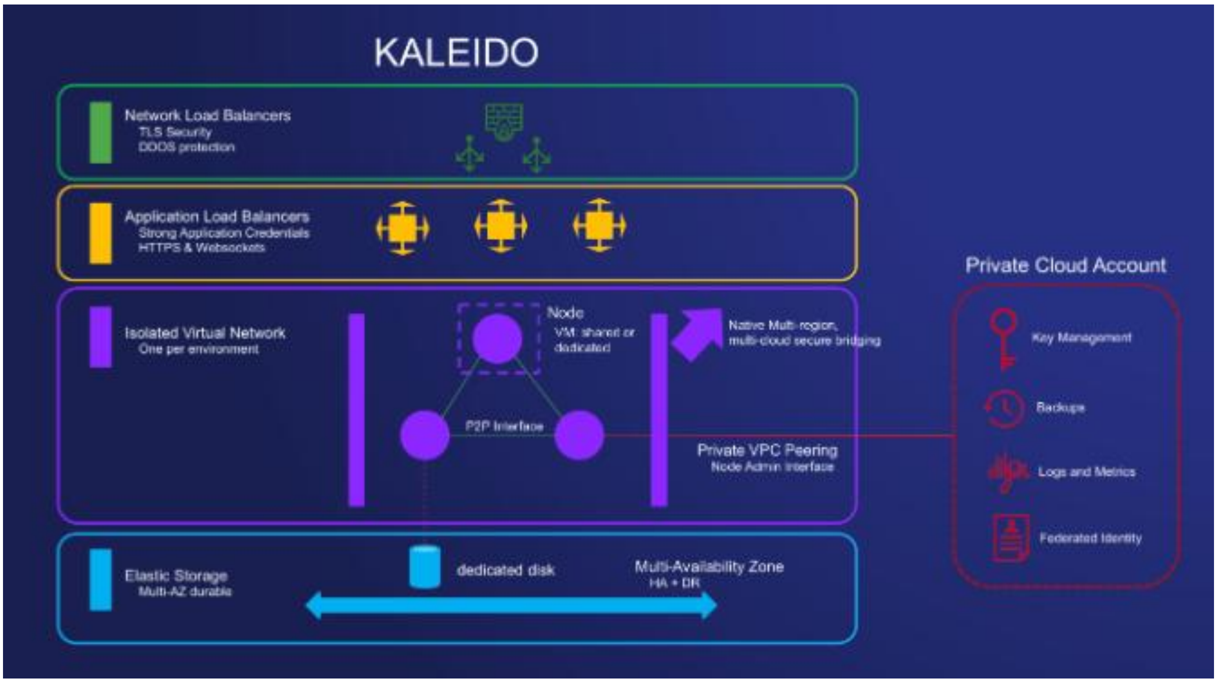
The proofs are usually expensive to calculate, but much, much cheaper to verify. Which means ZKPs are being considered as potential solutions for scaling throughput of high volume blockchain solutions, as well as providing privacy solutions. This is an exciting and fast evolving field, and choosing and implementing the correct algorithm needs some care to ensure that the required level of security is obtained.

Off-Chain Document Store

Use the document store service to securely upload and transfer files within a Kaleido environment. With blockchains being poorly suited for large payloads and sensitive data such as PII, an ancillary file sharing utility that can exist next to the chain provides immense benefit at both a business and security level. Transactions simply need to reference a file's hash, rather than dealing with the file contents directly. Any uploaded or received file can be conveniently organized using intuitive folder structures, and persisted in either Kaleido hosted storage or cloud-delegated and user-controlled services such as AWS S3 Buckets or Azure Blob containers. All transferred data is deterministically hashed, signed, compressed and asymmetrically encrypted in flight using proven public key infrastructure techniques, offering provable guarantees that only the intended recipient can effectively decrypt the packet. The service leverages the On-Chain Registry service for address look up and certificate functionality, as well as the Kaleido network's Kafka backbone for high throughput, fault tolerant reliable delivery and transport. Interaction with service is flexible, with support for the in-console graphical user interface, Kaleido RESTful APIs and socket.io connections for reliable event notification.

Tenancy & Isolation

Kaleido has robust processes implemented to ensure the proper protection of critical key materials, secure virtual networking, strong authentication schemes, highly available runtime components, firewalls and logical isolation, encrypted transport, cross-cloud tunneling and disaster recovery. The content in this section will explore these security and cloud architectural components, with an aim to address the core concerns of IT Ops professionals and organizational risk officers.



Network Layer

- Cloud native load balancers - ELB (elastic load balancers) and NLB (network load balancers)
- Serves a CA-signed and validated certificate for identity and validation
- TLS secured allowing for connection over HTTPS
- Default DDoS protection via the cloud native infrastructure services

Application Layer

- Connections are ultimately targeting the node ingress
- Nginx utilized for HTTPS calls
- HA Proxy utilized for WebSocket calls
- Application Credentials (strongly generated username password) combinations authenticate access to the ingress.
- Kaleido does not persist plaintext username secrets, rather a salted hash is kept and used for verification
- Verified calls are granted access to the isolated virtual network encapsulating the blockchain layer

Environments and Nodes

- Logical isolation per Kaleido network (environment)
- Network policies are used to isolate multi-tenant virtual environments across a shared VM pool
- Support for fully-dedicated VMs is available upon request to Kaleido
- Nodes are able to peer and gossip securely within their isolated environment
- Firewalls ensure no data leakage across isolated environments
- Resources are confined solely to the environment in which they are created. For example, application credentials in Environment A are unusable in Environment B
- Option for secure integration with user-controlled native cloud services. File system and key encryption, log streaming, file system backups and private connections.
- REST API Gateway available for idempotent at-least once delivery transaction submission via Kafka
- Event streams on a per-method subscription within smart contracts can send customized event payloads via Kafka to enumerated URLs

Storage

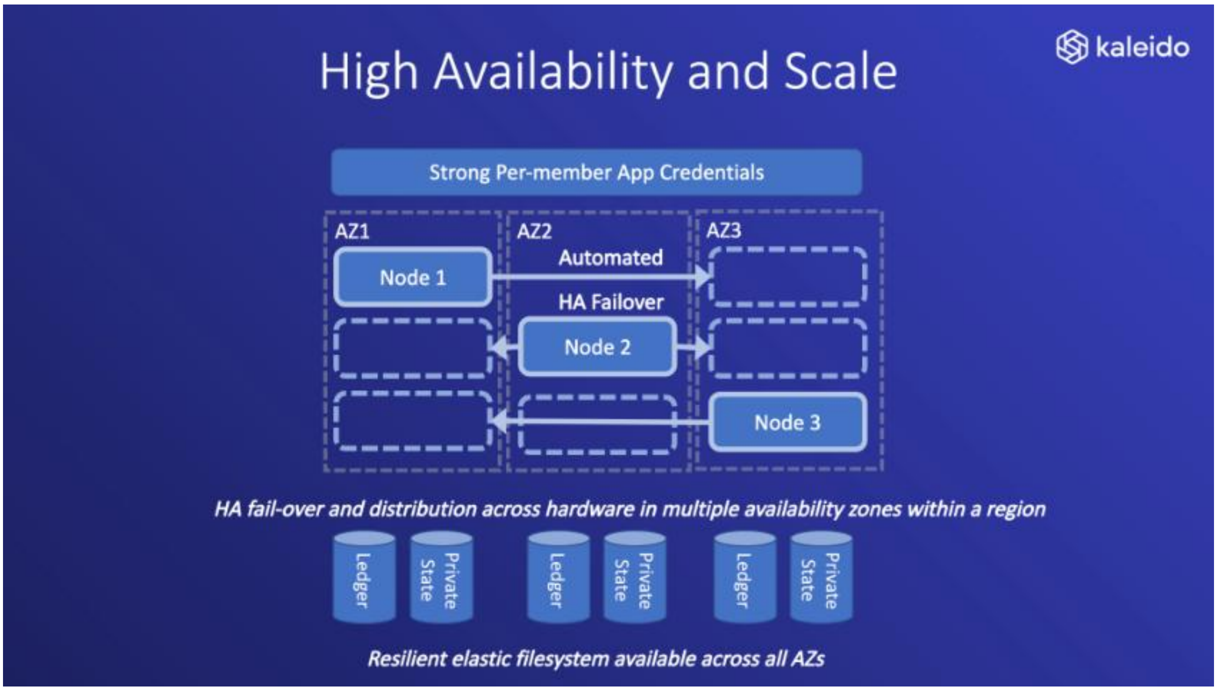
- Elastic File System instances are mounted on nodes to support on-demand scaling
- The cloud native file systems come with default High Availability and Disaster Recovery safeguards built in
- Dedicated directories of the elastic file systems are mounted and provisioned to each node ensuring data isolation in a shared virtual environment
- KMS integrations can be used to further encrypt anything written or mounted in the filesystem

Key Management

- Key materials generated upon node initialization
- Never leave the container in which they manifest
- AES-256 encrypted at rest
- EthWallet supports integration with cloud HSMs for transaction signing. Mandate is cloud provider support for the Ethereum curve - secp256k1
- Master encryption keys can be integrated with nodes upon creation for additional encryption of node file system and key materials. AWS KMS and Azure Key Vault
- No plaintext key material persistence for KMS integrated nodes. Cipher text stored on filesystem and decrypted material held in memory only
- All API calls accessing a user-owned encryption key are logged by the cloud provider and fully auditable
- All file systems AES-256 encrypted at rest
- For data in transit - HTTPS/WSS/Kafka - TLS 1.2 negotiable encryption is implemented
- Client side calls targeting a node or service ingress are TLS secured with strongly generated 256 bit security credentials.
- Kaleido uses salt hash verification against supplied application credentials to authenticate any inbound calls to the network; plaintext password is never persisted by Kaleido
- VPC Private Link can be configured to target a node's optional private ingress, keeping all traffic streams solely on AWS backbone

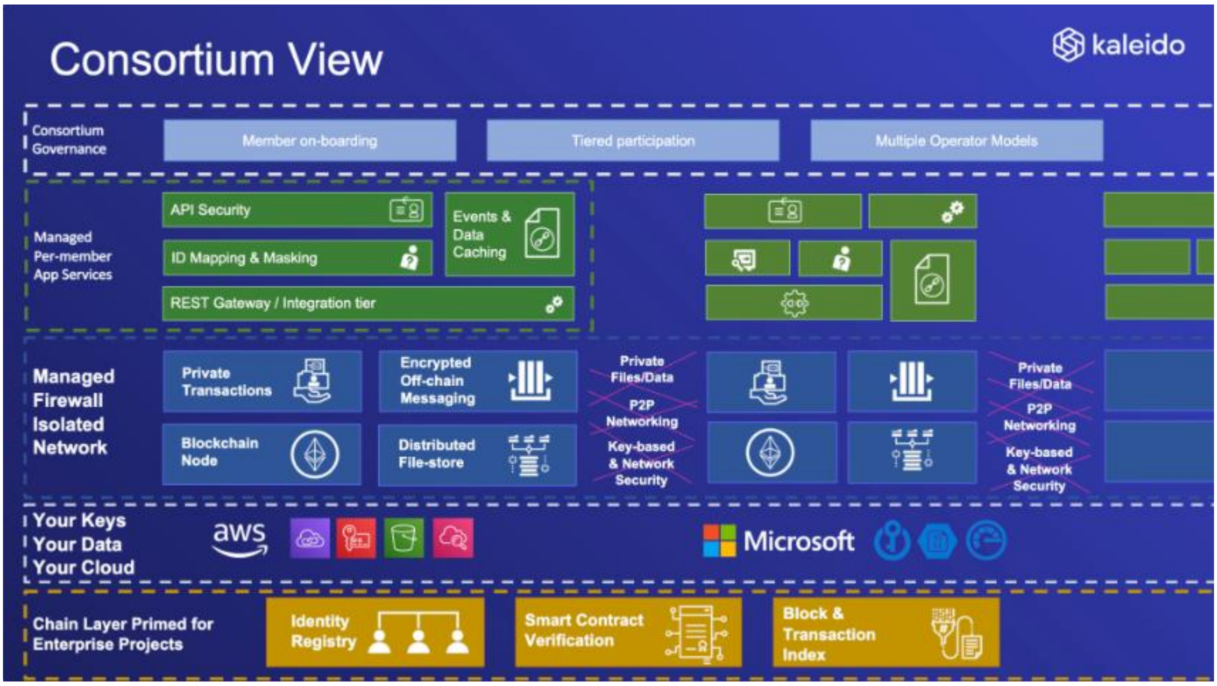
High Availability

- Kubernetes-based microservice architecture
- autoscaling Kubernetes clusters across multiple cloud region availability zones
- automatic failover with active/passive availability safeguards
- EFS supported disaster recover and high availability
- On demand backup to cloud storage services such as AWS S3 Buckets and Azure Blob containers
- ELB and NLB DDoS protections served natively via cloud services
- AWS Shield and AWS Route 53
- Next generation elastic filesystems for on-demand scaling
- Multi-region support for networks across different cloud regions and underlying AZs
- Hybrid support for networks across different cloud providers, available regions and underlying AZs


High Availability and Scale
kaleido
Strong Per-member App Credentials
AZ1
AZ2
AZ3
Node 1
Automated
HA Failover
Node 2
Node 3
HA fail-over and distribution across hardware in multiple availability zones within a region
Ledger
Private State
Ledger
Private State
Ledger
Private State
Resilient elastic filesystem available across all AZs

Decentralization Model

- Simplified email-based member on-boarding with configurable permissions.
- Operate as a centralized consortium with a single organization owning and operating network resources, or as decentralized with shared ownership across multiple entities.
- Organizations with different tiered subscriptions can coexist in the same consortium.
- Member services are protected with basic authentication credentials and controlled unilaterally by an owning member. These include REST Gateway resources, HD wallets, IPFS nodes and more.
- A consortium can host multiple blockchain networks (i.e. environments) with firewall isolation and environment-specific resources. No cross chatter or data leakage.
- Private transaction support via Quorum and Hyperledger Besu clients with private state information residing in a separate Patricia state trie. Non-privy participants will be unable to decrypt transaction payload for execution.
- Support for app to app messaging with asymmetrically encrypted payloads sent through Kaleido backbone; outbound messages are encrypted with counterpart's digital certificate
- Nodes, services and security credentials confined solely to their environment.
- Optional node and service integrations available with native cloud services such as Key Management Stores and Backup utilities.
- Organizational identities backed by digital certificates woven into chain layer with Registry Service.
- Block explorer supports granular inspection of transactions, blocks and smart contracts, with source code verification supported.



Privacy & Anonymity
Just because a Blockchain is private & permissioned, does not mean that the parties involved wish to make all of their data and transactions visible to all other participants within the business network. There are many reasons why data might need to be kept private.

Competition
Blockchain based business networks often foster *coopetition*. Enterprises collaborate in the business network for a common benefit, while actively competing with each other. These are some of the healthiest and most dynamic ecosystems. However, knowledge of details of a business transaction/trade, or the simple fact that a trade happened between a set of parties in the business network, might provide data that other parties could use to gain a competitive advantage.

Data Privacy
Some data is simply too sensitive to include on an immutable shared ledger. Consider the example of Sensitive Personal Information - this is often governed by strong data protection legislation such as GDPR, including controls that mean it must be deleted upon request from the owner of that data. Even if their data itself is not held on-chain there might be the potential for *metadata leakage* breaching privacy. Consider tracing drivers' license information on-chain - even if the identity is not widely available, you might be able to infer from direct knowledge of one citation a link to the token that represents a person on-chain. From there you can see that person's full history.

Privacy Solutions
Practical solutions for privacy are available today, and there are maturing solutions that are gaining traction and evolving to become practical for Enterprise use.

Proof-only Solutions and HD Wallets
In cases where all the material actions for a transaction are coordinated off-chain (see the next topic on off-chain comms), it might only be required to put a record of agreement on-chain. A signature from each party involved that agreement has been reached. This can take the form of a simple proof readable only to the other parties involved.

- Sign some data that designates the agreement using your private key
- Encrypt that data with the public key of each party that needs to read the proof
- Put each of those encrypted proofs on-chain
- Once all parties have submitted their proofs on-chain, the agreement is immutably recorded

However, in order to avoid leaking information about who is involved in a transaction you must first mask the identity that is submitting the transaction to the chain. It is possible in some cases to simply generate a completely random address to sign the Blockchain transaction. However, that does not allow for permissioning of who can submit signatures to the contract, or traceability back to who submitted the transaction. As such, a Hierarchically Deterministic (HD) Wallet is commonly used to generate a single-use Ethereum address that the owner can prove is their own if asked.

Private Transaction Managers & Enclaves

One of the most popular options to make transactions and data visible only to a subset of the participants, is to use a Private Transaction Manager such as Tessera from Quorum (the successor to the original Constellation project in this space).

These work by sending off-chain point-to-point encrypted communications to select parties that are allowed to view the full input data for the transactions. These select parties can process the full transaction, and update their state to include the full results. Other parties in the network maintain a proof that the data was sent, a hash of the payload and the address of the sender, but cannot execute the transaction as they never receive the encrypted payload.

Again a HD Wallet can be combined with private transaction managers to submit the transactions, if knowledge of which parties are transacting is itself a privacy concern.

Key aspects of this approach that differentiate it from roll-your-own off-chain communications are:

- Agreed Smart Contract logic is used, just like any other transaction
- Installation of the Smart Contract logic is itself a private transaction
- The deterministic properties of the EVM are maintained - parties can compare merkle roots of their contract instances to confirm they have the same state
- The state and events of the Smart Contracts can be processed just like any other Ethereum transaction
- Private Smart Contracts have read-only access to the all-participant chain