

N E W N A L

Personal Data Node

Newnal's data management architecture:
build services on a wider range of customer-permitted personal data,
without collecting or storing it yourself

This document explains what a Personal Data Node is and how it is structured, how data is shared through Data Spheres,
what enterprises gain, and the role of the public ledger (InfraBlockchain).

NEWNAL Inc. · 2026

Contents

Overview	3
Four problems enterprises face	3
Newnal's approach	3
What enterprises gain	3
Track record	3
1. Terminology	4
2. Personal Data Node	5
2.1 Definition	5
2.2 What "the customer owns it" means	5
2.3 Node components	5
3. Data Spheres and the Service Agent Node	6
3.1 Data Sphere	6
3.2 Service Agent Node	7
3.3 Overall architecture and data flow	8
3.4 What changes for the company	8
4. Identity and keys	9
5. The public ledger — what it records, and why InfraBlockchain	9
5.1 What is recorded on the ledger	9
5.2 Why a public ledger, and why InfraBlockchain	10
6. Security	10
7. Partner roles and adoption	11
7.1 Adoption path	12
7.2 Replacing existing member functions	12
8. Existing infrastructure and regulation	12
9. Frequently asked questions	13

Overview

Four problems enterprises face

Problem	Description
① Personalization needs personal data	For AI to know a customer, it needs that customer's data. Without data, AI gives every customer the same answer.
② Collecting data creates liability	The moment customer data is gathered on company servers, responsibility for storage, breaches, and regulatory compliance follows. Insiders with legitimate access and attackers with stolen credentials are hard to stop with security spending alone.
③ Your own data is not enough	A company's servers hold only the data generated by its own service. No single company can build a service that understands a customer's health, daily life, and spending together, and receiving data through partnerships brings back the liability of transfer and storage.
④ Every company builds the same infrastructure	Collection pipelines, storage, security staff, consent management, and breach response are built separately by each company. The same infrastructure is duplicated as many times as there are companies, for the same customer's data.

All four problems share one cause: data is aggregated **per company**. In a company-centric structure, the more data you collect for personalization, the larger your storage liability grows.

Newnal's approach

Newnal changes the unit of aggregation from the company to the **individual**. Every customer has a **Personal Data Node**, a per-person cloud server, and the customer's data accumulates there. A company proposes a scope of sharing called a **Data Sphere**, and only the data in spheres the customer has permitted is synchronized to the company's **Service Agent Node**. Service features and AI agents run on the Service Agent Node, and the company's existing systems, web, and apps receive the results through APIs. Both nodes run published open-source code inside confidential computing environments, so even the party operating the cloud infrastructure (including Newnal) cannot access personal data while it is being processed. Verification records of which node permitted what and who accessed it when are written to a public ledger outside Newnal (InfraBlockchain).

What enterprises gain

First, services are built on all the data the customer permits, not just the data you collected. Your existing data remains, and the customer-permitted portion of data recorded in the node by other services, institutions, and devices is added. This is a scope no company can reach by collecting data on its own.

Second, personal-data collection and storage infrastructure is lifted out of your existing systems. Collection, storage, consent management, and access control for personal data are handled by the Personal Data Node and the Service Agent Node. Your existing systems never connect to Personal Data Nodes directly; they only call the Service Agent Node's APIs. The Service Agent Node is either built and operated by Newnal as a managed service, or built by the company itself with Newnal's open framework.

Track record

COOV, the vaccine pass developed and operated by Newnal, was used by 43 million people in the Republic of Korea, verified up to 30 million credentials per day, and issued 260 million DIDs. Original records stayed on the issuing institutions' servers; only verifiable credentials were stored on personal devices. What COOV proved at national scale is DID, verifiable credentials, and the public ledger. The Personal Data Node and the Service Agent

Node are components added on top, and are in operation today in Newnal aios. Newnal holds more than 90 registered and pending patents.

1. Terminology

Term	Meaning
Personal Data Node	A per-person cloud server assigned to each customer. It stores the customer's data encrypted, provides data only within permitted sphere scopes, and keeps access records. In this document, "node" means the Personal Data Node.
Service Agent Node	A server holding access rights to a sphere. It synchronizes records from the nodes of customers who joined the sphere, builds a data view for the service, runs service features and AI agents, and exposes them through APIs.
Data Sphere	A scope of sharing defined by a sphere owner: a data schema and a purpose. When a customer agrees to join, a repository for that sphere is created inside the customer's node. A company may operate several spheres for different purposes.
Sphere Owner	The party that opens a sphere, defines its data schema, API contract, and participation criteria, and issues Sphere Access Credentials to Service Agent Nodes. A company can be a sphere owner.
Sphere Access Credential	A credential issued by a sphere owner to a Service Agent Node. A node verifies it before providing data, and provides only the data of that sphere.
Data View	The data a Service Agent Node assembles from participating customers' sphere records, structured for the service. It occupies the place that a customer database held in the conventional model.
DID	A decentralized identifier that refers to a person or service in place of a real name or a company account. The DID's key list and the node's access address are registered in the DID Document on the public ledger.
Verifiable Credential	A credential issued to an individual by an institution or company, whose authenticity anyone can verify. The original stays with the issuer; only the credential is stored in the node. This is distinct from the Sphere Access Credential issued to Service Agent Nodes.
Public ledger (InfraBlockchain)	A public blockchain that records DID Documents, sphere definitions, settlement tokens, and verification information for access records. It operates without issuing a cryptocurrency and never stores original data.
Confidential computing environment	An execution environment in which even memory is encrypted, so cloud administrators cannot read its contents through software access. Whether the running code matches the published code can be verified externally (remote attestation).
Reverse Login	A flow in which a service requests to join a customer's node sphere and the customer approves. The direction is the opposite of the conventional flow, where the customer logs in to a service and hands over data.

2. Personal Data Node

2.1 Definition

A Personal Data Node is a **per-person cloud server** assigned to each customer. The node stores the customer's data encrypted, provides data to services only within the scope of Data Spheres the customer has permitted, and records every access. The node runs in a confidential computing environment and is registered on the public ledger under the customer's DID, not a company account. Control of the node belongs to the customer.

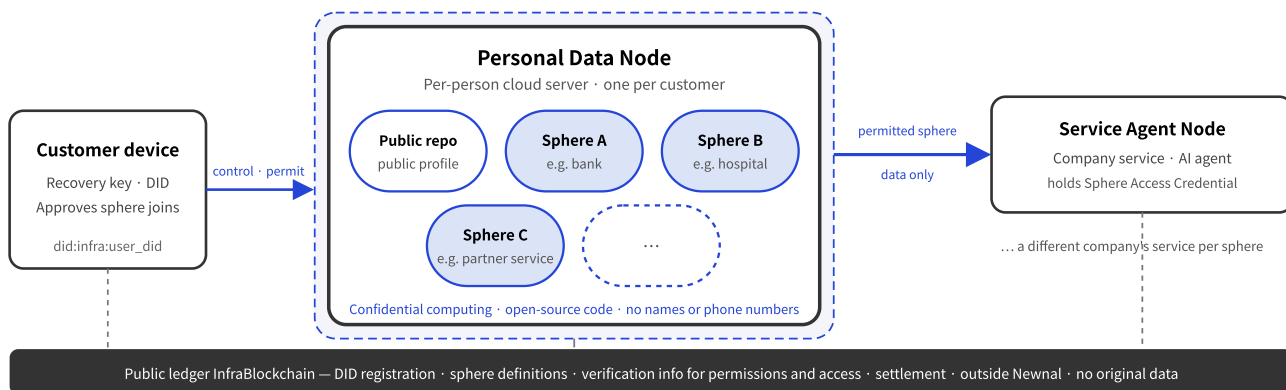


Figure 1. The customer's device controls the node and approves sphere participation; the node provides only permitted sphere data to the corresponding service. Verification information remains on a public ledger outside Newnal.

A node is not a dedicated physical server per customer. Many logically isolated nodes run on one physical server, with data repositories, access policies, access records, and settlement separated per DID. The node is not a container that data is poured into; it is a server that acts under the customer's DID, and databases and cloud are components the node uses.

2.2 What "the customer owns it" means

The node runs in the cloud; the customer's device holds only the recovery key and a small amount of private data. In this document, ownership does not mean storage location. It means the following three conditions.

Condition	Description
Registration	The node is registered on the public ledger under the customer's DID. It belongs to no company's account system.
Keys	The top-level recovery key that can rotate or revoke the node's signing key and data key belongs to the customer. Newnal, the cloud operator, and partner companies do not have this authority.
Portability	By changing the node's access address registered on the public ledger, the customer can move the node to another cloud or operator. Node location can be set per jurisdiction.

2.3 Node components

Component	Description	Technology
Storage	Customer data is stored encrypted with a per-node data key. Storage is divided into a public repository and per-sphere repositories. How the data key is held and released is described in Section 4.	Per-node data key

Component	Description	Technology
Execution	The published node server code runs in a confidential computing environment. Cloud administrators cannot read its contents, and whether the running code matches the published code can be verified externally.	Confidential computing, remote attestation, open source
Identity	The node is registered under a DID, not a real name or account. Because identification, contact, and authentication are handled by the DID, the node's system identifiers contain no real names or phone numbers.	DID, DID Document
Keys	The signing key and the data key are held in a server-side HSM; the recovery key that can rotate both is on the customer's device. Details are in Section 4.	HSM, device secure element, multi-key DID
Access control	Data within the customer-permitted sphere scope is provided only to Service Agent Nodes that pass verification. If permission is withdrawn, synchronization stops.	Sphere definition (public ledger), attestation verification
Records	Permissions and access are recorded in the node and reported to the customer. Verification information for the records is anchored to the public ledger.	Node log, public-ledger anchoring
Receiving verifiable credentials	Verifiable credentials issued to the individual by hospitals, institutions, and companies are delivered to the node. Originals stay with the issuer.	Verifiable Credential (COOV model)
Communication	Communication among the node, the device, and Service Agent Nodes is end-to-end encrypted and mutually authenticated by DID.	DID-based end-to-end encryption
Settlement	Which sphere used which data is aggregated per DID, and fees and revenue are settled accordingly.	Public-ledger token contract

This architecture holds only when four things are present together in a single node. Without **confidential execution and published code**, the operator can read the contents and the node is just a company server by another name. Without **the recovery key in the customer's hands**, the operator who holds the signing key and data key takes the place of the node's owner. Without **DID registration**, the node is tied to a real name or a company account and becomes one company's asset. Without **a public ledger outside Newnal**, the claim that records have not been altered rests on Newnal's word alone. Newnal operates these four conditions as a single architecture.

3. Data Spheres and the Service Agent Node

3.1 Data Sphere

A Data Sphere is a scope of sharing defined by a sphere owner. The owner sets the sphere's data schema, API contract, and participation criteria and registers them on the public ledger. When a customer agrees to join, a repository for that sphere is created inside the customer's node, and the customer's node and the service write records there. **A company can create multiple Data Spheres to build its services.** Even for a single service, spheres can be separated by purpose, such as a profile sphere, an activity sphere, and a payment sphere, and customers grant and withdraw permission sphere by sphere.

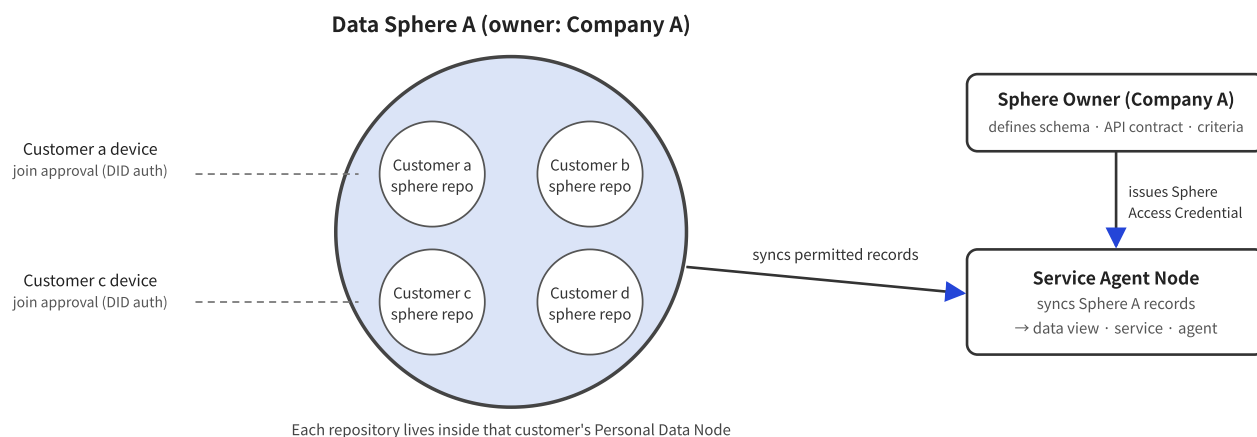


Figure 2. A Data Sphere is a sharing scope that spans many customers' nodes. Only a Service Agent Node holding an access credential issued by the sphere owner can synchronize records within that scope.

3.2 Service Agent Node

The Service Agent Node is the server that obtains access rights from the sphere owner, **synchronizes** records from the Personal Data Nodes participating in the sphere, builds the **data view** needed to implement the service, and **runs** the actual service features and AI agents. Service features are provided as APIs to the company's internal systems and web/app clients.

Services and agents do not run inside the Personal Data Node. Data from spheres the customer has permitted is shared from the Personal Data Node to the Service Agent Node and processed there. For that reason, the Service Agent Node is protected to the same standard as the Personal Data Node.

Before sending records, the Personal Data Node checks two things. ① The Sphere Access Credential issued by the sphere owner. ② Remote attestation that the Service Agent Node is running the published code in a confidential computing environment. If either check fails, the node sends no data. This verification applies equally to Service Agent Nodes operated by Newnal and to those built by companies themselves. If a customer withdraws permission, synchronization stops and that customer's records are removed from the data view.

A Service Agent Node can be provisioned in two ways. **Newnal managed service:** Newnal builds and operates the Service Agent Node in a confidential computing environment, and the company defines only the data view and service logic. **Self-hosted by the company:** the company builds it in its own cloud using Newnal's open framework. In this case the company is responsible for operating the environment, not for access to the data. Either way, the company's existing systems never connect to Personal Data Nodes directly and only call the Service Agent Node's APIs, so no personal-data collection or management system is needed in the existing systems.

3.3 Overall architecture and data flow

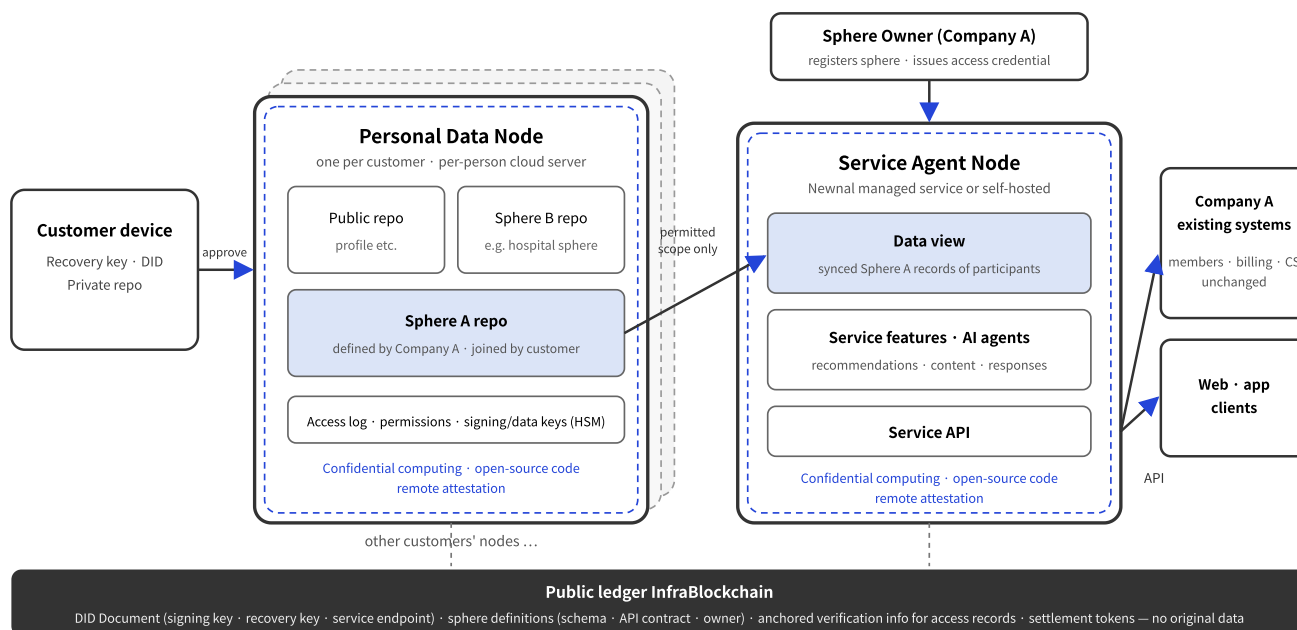
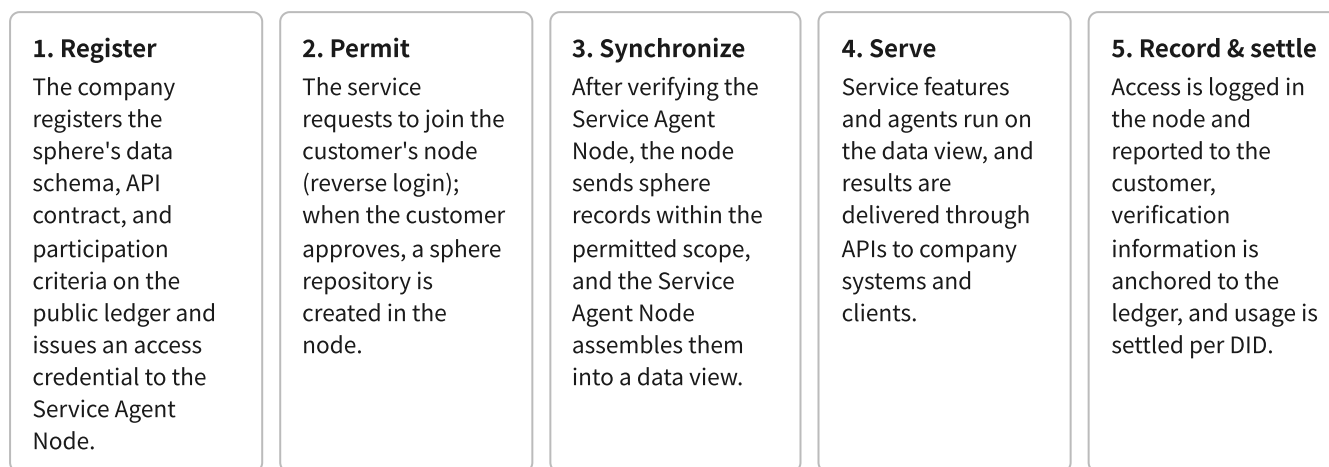


Figure 3. The customer approves sphere participation on the device; the Service Agent Node synchronizes only permitted sphere records and runs the service; the company's existing systems and clients receive results through APIs. Verification information remains on a public ledger outside Newnal.



3.4 What changes for the company

Where the customer database used to sit, there is now the Service Agent Node's data view. The data view consists of the data of customers who joined the spheres the company defined. It contains two kinds of data. One is the portion of the company's existing data that the company has written into sphere repositories inside customers' nodes. The other is the data recorded in nodes by other services, institutions, and devices that customers have additionally permitted for this sphere. Existing database records are not part of the data view until they are written into sphere repositories, and the company migrates only what each service needs. Because personal data lives in the Service Agent Node rather than in the company's existing systems, those systems collect and store no personal data and handle only API results. The company's customer database does not disappear; it becomes a protected data view, and the scope of data expands.

4. Identity and keys

In short: neither Newnal nor the cloud operator can open a node's data, and only the customer has the authority to change who owns the node. What makes this possible is where the keys are placed. A single DID has several keys with different authorities, and the key list and the node's access address are recorded in the DID Document on the public ledger. The signing key and the data key reside in a server-side HSM (Hardware Security Module) where the node runs; the recovery key that can rotate both resides with the customer.

Key	Location and role	What it guarantees
Signing key	The key the node uses to sign on the customer's behalf in the cloud. It is held in the server-side HSM, never exported, and used only for signing.	The node operates without the device present. The raw key is never exposed in server memory or on disk.
Data key	A per-node key that encrypts the node's stored data. Like the signing key it is held by the server-side HSM, and actual record encryption and decryption are performed inside the node's execution environment with this key. The HSM releases the key only to node execution environments that pass remote attestation; administrator privileges cannot extract it. The customer can rotate the data key with the recovery key.	Stored data is decrypted only inside an attested node execution environment. A copy of the disk reveals nothing, and the customer can rotate the key if it is ever in doubt.
Recovery key	The top-level key that can rotate or revoke the signing key and data key the node is using. It is stored in the secure element of the customer's device and can be backed up by the customer. Several can be created, and some can be entrusted to a trusted recovery-key custody service.	Only the customer can change who owns the node. Switching operators and moving the node are possible.
Recovery-key custody service	Holds recovery keys entrusted by customers, isolated per customer, and provides the recovery procedure only to customers who pass the identity verification configured at enrollment. Administrator privileges cannot read the keys.	A DID can be recovered even if both the device and the backup are lost. No single intrusion can yield many customers' keys.

If the device is lost, the node keeps running in the cloud. The customer registers a recovery key on a new device from a backup or from the custody service, and rotates the signing key and data key if needed. For minors, a guardian takes part in recovery.

5. The public ledger — what it records, and why InfraBlockchain

5.1 What is recorded on the ledger

Record	Written directly to the ledger	Kept in the node; only verification info anchored
Registration (whose node)	DID Document: signing keys, recovery keys, service endpoint	—
Sphere definitions	Owner DID, data schema, API contract, participation criteria	—

Record	Written directly to the ledger	Kept in the node; only verification info anchored
Permissions and access	—	Which spheres were joined and who synchronized what and when are recorded in the node, and a hash of those records is periodically anchored to the ledger.
Settlement	Token contract: transfer of fees and revenue	Per-sphere usage aggregation is computed in the node.

There are two reasons the raw access records are not put on the ledger. Which DID is connected to which service is itself sensitive, and recording every access event for tens of millions of people is beyond any ledger. The anchored verification information alone is enough to confirm, without going through Newnal, that records have not been altered after the fact.

5.2 Why a public ledger, and why InfraBlockchain

Registrations, sphere definitions, and verification information must satisfy three conditions. Newnal alone must not be able to change them; customers, companies, and auditors must be able to verify them without going through Newnal; and multiple companies that do not trust one another must share the same records. A database owned by Newnal fails the first two; a ledger kept by any one company fails the third.

InfraBlockchain property	Description	Result
No cryptocurrency	The network runs without a native cryptocurrency. Fees are paid in tokens pegged to fiat currency.	Fees are stable, and governments and financial institutions can use it. This was a condition that allowed the Korean government to adopt COOV.
Public	Anyone can look up and verify the records.	Customers, companies, auditors, and regulators confirm records without going through Newnal.
Proof-of-Transaction	Block producers are chosen by actual service transaction volume processed, not by mining or token holdings. Block producers are multiple identified service providers.	Accountability is clear, there is no mining-related power consumption, and Newnal alone cannot alter the records.
No original data	The ledger holds only registrations, definitions, verification information, and settlement.	No conflict with the right to erasure, and a leaked ledger contains no data.
Proven at national scale	The ledger operated a national-scale service for COOV (see Track record in the Overview).	Performance and stability are demonstrated by an operating record.

6. Security

Attack path	Company-server model	Personal Data Node + Service Agent Node
Insider leaks with legitimate privileges	Legitimate privileges allow querying the entire customer dataset, so security systems cannot block it.	No operator privilege can read customer data. Newnal's administrator privilege is the authority to operate nodes, not to read their contents.

Attack path	Company-server model	Personal Data Node + Service Agent Node
Attacker steals an employee account	The stolen privileges give access to the database.	There is no account with read access to steal. Even partner-company staff cannot reach data outside the sphere scope, because it does not exist in the Service Agent Node.
Attacker breaks into a server	Millions of records are copied at once, together with real names and contact details.	Data is encrypted per node and the decryption key never leaves the server. System identifiers are DIDs, and the Service Agent Node holds only the data of its sphere scope.
Cloud administrator inspects the server	A top-privilege administrator can read memory and disk.	Software access cannot read the contents, and a disk copy reveals nothing (Section 4). Physical attacks on live memory remain; Newnal applies hardware vendors' security updates and limits the scope each execution environment handles to reduce impact.
Operator changes code without disclosure	Undetectable from outside.	The node server code is published, and evidence that the running code matches it (remote attestation) is verified by third parties using the hardware vendor's attestation scheme.
Partner service reaches beyond its scope	Permitted scope is kept only as a promise.	Sphere definitions are on the public ledger, and the node does not deliver data outside the sphere named by the access credential. Access within scope is also logged and reported.

Newnal's approach to breaches is not to make intrusion impossible. It is to make large-scale extraction hard, and to ensure that whatever is extracted cannot be linked to a person by anything other than the DID system identifier. If a company places names or contact details inside a sphere, that data is protected within the sphere scope but does not become unidentifiable. For this reason, Newnal recommends that highly sensitive data not be stored in Data Spheres wherever possible.

Three risks remain. A targeted attack on a specific customer's device could expose that customer's recovery key. The availability and operation of a Service Agent Node that a company hosts itself are the company's responsibility (data access control applies identically through the verification in 3.2). Payment information held by payment processors and sales channels is outside the node architecture.

7. Partner roles and adoption

Partner's role	Not the partner's responsibility	Partner's benefit
Service planning and operation; brand, sales, pricing, product mix	Building personal-data collection and storage infrastructure	Personalized services built on all the data the customer permits
Sphere design and registration (with Newnal's help)	Building login, identity verification, and notification infrastructure	No personal data accumulating in existing systems
Data view and service logic; design of content, recommendations, and AI responses	Breach response and security staffing (with the managed service)	No need to acquire data before launching a service

Partner's role	Not the partner's responsibility	Partner's benefit
Screens and wording that ask customers for permission	Building consent management and access-record systems	Automatic per-DID settlement; sales and subscription revenue
Payments (through existing channels)	—	Start by registering a sphere on nodes other companies have already connected

7.1 Adoption path

1. Design spheres

Divide the data the new service needs into purpose-specific spheres, define schemas and API contracts, and register them on the public ledger. Newnal works with you on the design.

2. Set up the Service Agent Node

Subscribe to Newnal's managed service or self-host with the open framework. Implement the data view and service logic and define the APIs.

3. Connect existing systems

Issue existing members a verifiable credential stating "this is our member" to link them to their nodes, and write any existing data the new service needs into those customers' sphere repositories. Existing systems call the Service Agent Node's APIs. Membership, billing, and customer service remain as they are.

These three steps are enough to launch a new service. The scope and timing of migrating personal data from existing databases into sphere repositories, and whether to delete server copies afterwards, are decided by the company service by service, on its own schedule. Data that is not migrated is not part of the data view; for every record migrated, storage liability and infrastructure cost in existing systems decrease.

7.2 Replacing existing member functions

Function	Conventional method	Node-architecture method
Login · identity check	ID, password, phone verification	Proven by DID and device. Existing members are linked by a verifiable credential.
Contact · notifications	Phone number, email, app push	Encrypted communication mutually authenticated by DID. Only parties the customer has permitted can connect.
Subscription · entitlement	Billing history tied to an account	Only a verifiable credential stating "this DID's subscription is valid" is verified.
Personalization · recommendations	Joined with server-side user profiles	Joined in the Service Agent Node's data view with data in the permitted scope.
Payment · refund · CS	Account lookup	Existing sales channels and payment processors continue to handle these.
Loss · recovery	Identity check by national ID or phone number	A customer-driven recovery-key procedure. For minors, a guardian takes part.

8. Existing infrastructure and regulation

Nodes do not replace the cloud, AI models, or operating systems. The cloud runs nodes; AI models operate inside Service Agent Nodes with context limited to the permitted scope; and Newnal aios is deployed as the operating

system on dedicated devices, as a layer on existing devices, and as a node execution layer for services without a device. As the use of node-based services and AI agents grows, so does the consumption of cloud and AI models. Newnal is a customer of existing infrastructure providers, not a competitor.

Regulatory requirement	Response of the node architecture
Data minimization (GDPR Art. 5)	Real names and contact details are not collected as system identifiers, and only the data needed is shared sphere by sphere.
Withdrawal as easy as consent (Art. 7)	Withdrawing a sphere stops synchronization and removes the data from the data view.
Privacy by design (Art. 25)	Access control is built into the architecture.
Right to erasure (Art. 17) · portability (Art. 20)	Originals are in the customer's node, and the customer can move the entire node to another cloud.
Right of access (Art. 15)	Every access is recorded and reported to the data subject.
Restrictions on international transfer (Chapter V)	Nodes and Service Agent Nodes can be located per jurisdiction.
Protection of children (Art. 8 · COPPA)	Children's identifiers and contact details are not collected; the guardian is the party for payment and consent.

This table is not legal advice. Jurisdiction-specific application, including the partner company's status as controller or processor, is determined together with the partner's legal review; Newnal provides the technical documentation required.

9. Frequently asked questions

Q1. How is a node different from a database? A database controls access by the owning company's accounts. A node controls access by the customer's DID and sphere permissions, and never delivers data outside the permitted scope. Databases and cloud are components used inside the node.

Q2. Without a member database, how do we know customer counts and active users? Everyone who has joined the company's sphere is the company's customer. Customer counts, activity, and subscription status are aggregated per DID in the Service Agent Node. Because the unit of aggregation is the DID, no list identifying customers is created unless the company includes real names or contact details in the sphere.

Q3. Millions of customers means millions of nodes — what about cost? A node is not a dedicated server per customer; it is a logically isolated cloud server inside a confidential VM. Cost scales with active customers, and the spending that companies previously devoted to storage, security, and breach response in their existing systems decreases.

Q4. If data gathers in the Service Agent Node, isn't that just a company server? Three things differ. Only data the customer permitted, sphere by sphere, is present. Operators cannot read it. Withdrawal removes it. And no personal data accumulates in the company's existing systems.

Q5. As the sphere owner, can a company read all participating customers' data? A sphere owner's authority is to define the sphere and manage participation criteria and access credentials. Customer data is synchronized only to the Service Agent Node, only within the scope the customer permitted; there is no authority to read it through any other path.

Q6. What if a customer loses their device? The node keeps running in the cloud. The customer recovers the DID from a backed-up recovery key or from the custody service and registers a new device. For minors, a guardian takes part.

Q7. Who is responsible if something goes wrong? Operation of Personal Data Nodes, the public ledger, and Service Agent Nodes provided as a managed service is Newnal's responsibility, and Newnal demonstrates compliance through external audits. Operation of a Service Agent Node a company hosts itself is that company's responsibility.

Q8. What if law enforcement demands data? What Newnal can produce is encrypted storage and access records. Newnal's administrator privileges cannot extract the data key (Section 4), so Newnal cannot read and produce the contents. Whether to provide the contents is decided by the customer, the node's owner.

Q9. Isn't blockchain slow and expensive? The ledger records only registrations, definitions, verification information, and settlement; raw access records and data stay in the node. InfraBlockchain runs without a cryptocurrency, so fees are stable, and it operated at national scale for COOV.

Q10. What if a customer writes their own name in a story or review? The system cannot prevent that. Newnal's claim is not "identification is impossible" but "no real name, contact detail, or account that links records to an actual person exists among the system identifiers."

Contact Partnership inquiries: partner@newnal.com

Customer data lives in the customer's Personal Data Node. Partner companies build services on the Data Spheres customers have permitted and keep no personal data in their existing systems. Nodes and the public ledger are operated by Newnal, under Newnal's responsibility.