
The Operating System for the AI Era: Newnal aios

An operating system that runs and protects personal data in units of Personal Data Nodes,
and runs AI agents within the scope each person permits

Derives three conditions from the definition of an operating system,
and explains that Newnal aios is an operating system that meets them on a per-person basis.

Contents

Overview	3
Why a new operating system is needed	3
aios	3
Structure of this document	3
1. Definition of an Operating System	4
2. Existing Operating Systems — Applying the Definition and Its Limits	5
3. Composition of aios — Four Components	6
4. Managing per Person, Not per Device — The Personal Data Node	7
5. The Shift in Interface	8
5.1 Eras of computing defined by interface	8
5.2 Conditions for the AI Interface	9
5.3 Example — How a single proposal is processed	9
6. Applying the Definition to aios — How It Differs from Existing Operating Systems	10
7. Deployment Forms	11
8. Scope of Adoption — Phased Adoption	12
9. Anticipated Objections and Answers	13
Conclusion	14

Overview

Why a new operating system is needed

For an AI agent to work for a person, three things are needed. The person's data must be gathered in one place, the person must be able to decide how the data is used, and the agent must be able to propose first based on the data, without waiting for the person's command. Existing operating systems (iOS, Android, macOS, Windows) do these three things for files and apps inside the device, but not for a person's data or for AI agents. The moment an app takes data to the company's server, the operating system's management ends, and from then on the company, not the person, decides how the data is used.

This is not a flaw in existing operating systems. They were designed to manage resources inside the device and to run apps, and they still serve that purpose. What has changed is that the AI era has produced a new resource, personal data, and a new kind of program to run, AI agents, and these call for an operating system that manages them on a per-person basis.

aios

Newnal aios is an operating system that runs and protects personal data in units of Personal Data Nodes and runs AI agents in the Personal Data Node environment. aios works with or without a device, and on top of an existing operating system or without one. On a dedicated device it runs as an independent operating system that also manages the hardware; on a device with an existing operating system it runs as an operating-system layer on top of that system; and inside a service with no device it runs as an operating-system layer that runs nodes. In all three cases it does the same work.

For an operating system to handle a resource, the resource needs a unit of management. Existing operating systems manage storage in units of files and execution in units of processes. aios manages personal data, and execution on that data, in units of Personal Data Nodes. A node is a program that runs one per person, stores data, enforces permissions, and runs services. Just as an existing operating system creates, isolates, and protects processes, aios creates nodes (one per person), isolates them, protects them, and moves them. Because nodes exist, aios manages resources, controls access, and runs agents on a per-person basis. In aios, not only a person's data but also the new data that services produce from it accumulates in the person's node and, with the person's permission, can be used again by other services. Companies adopt aios in the scope they choose, from part of it (the data layer) to all of it.

Structure of this document

Chapter 1 derives three conditions from the textbook definition of an operating system. Chapter 2 compares the units in which existing operating systems carry out these conditions with the units AI agents require. Chapter 3 covers the four components of aios; Chapter 4, per-person management and the Personal Data Node; Chapter 5, the shift in interface and an example of how a single proposal

is processed; Chapter 6, the result of applying the definition to aios and how it differs from existing operating systems; Chapter 7, deployment forms; Chapter 8, the scope of phased adoption; and Chapter 9, anticipated objections and answers.

1. Definition of an Operating System

Operating system textbooks define the operating system as follows.

- Tanenbaum, Modern Operating Systems: An operating system plays two roles. As an extended machine, it hides the complex details of the hardware and gives programs clean abstractions (processes, files, and so on); as a resource manager, it allocates and protects the processor, memory, storage, and I/O devices so that multiple programs can share them.
- Silberschatz, Operating System Concepts: An operating system is a program that manages the computer hardware, provides a basis for application programs, and acts as an intermediary between the user and the hardware.

The two definitions share three conditions for an operating system.

Condition	Description
Resource management	The operating system allocates and protects resources so that multiple programs can share them. Resources include files and data as well as the processor, memory, and devices.
Access control	The operating system determines and enforces which programs and users can access which resources.
Execution base	The operating system provides the abstractions and interfaces on which application programs run, and it runs those programs.

What the definitions do not include is also worth stating. They do not require an operating system to boot the device with its own kernel, they do not limit the resources it manages to hardware, and they do not rule out running on top of other software. Android runs on the Linux kernel, yet it is called an operating system. The test is whether the three conditions are carried out.

The definitions require one more thing. A resource must have a **unit of management** that the operating system can handle. Existing operating systems manage storage in units of files and execution in units of processes. Because files and processes exist as units, the operating system can allocate resources, control access, and run programs. For an AI-era operating system to manage personal data as a resource, personal data also needs a unit of management. In Newnal aios that unit is the Personal Data Node, which stores data like a file and runs like a process.

2. Existing Operating Systems — Applying the Definition and Its Limits

Taking iOS and Android as examples of existing operating systems gives the following. macOS and Windows have the same structure.

Condition	iOS	Android
Resource management	Hardware of Apple devices and files on the device	Hardware of devices from many manufacturers and files on the device
Access control	App sandbox; user permission for device resources such as the camera and contacts	App sandbox; user permission for device resources
Execution base	Running apps	Running apps

The access control of existing operating systems reaches only as far as the resources inside the device. Once an app takes a user's data to the app company's server, that data lies outside the operating system's management and control. The same person's data is scattered across apps and stored on each company's server, and no operating system manages server-side data as a resource. This is because existing operating systems have no unit for handling a person's data. A file is a unit only inside the device; data that goes to a server becomes a row in the company's database. Likewise, data that a service produces by analyzing and processing a person's data stays within that service and is hard to reuse in other services.

Mapping the three needs from the overview (data in one place, scope of use set by the person, agents that act first) onto the three conditions of an operating system gives the following. Existing operating systems were designed to carry out all three conditions per device and per app, not per person.

Condition	Unit existing OSes work in	Unit AI agents need	What existing OSes lack
Resource management	Hardware and files inside the device	All of one person's data	One person's data is scattered across apps on each company's server, and the operating system does not manage server-side data.

Condition	Unit existing OSes work in	Unit AI agents need	What existing OSes lack
Access control	An app's permission to device resources	An agent's permission to a person's data	Once data leaves the device for a company server, the operating system's control ends. The company, not the person, decides the scope of use, and records of permission and access stay only inside the company.
Execution base	Apps that wait for the user's command	Agents that propose first based on data	An app sees only its own company's server data and cannot access the same person's data on other companies' servers.

There are three ways to build AI agents on existing operating systems: build an agent that sees only one app's data, have the person hand over data whenever it is needed, or gather data from multiple apps on one company's server. The first cannot know the whole person and so cannot deliver a tailored service; the second works only when the person gives a command and so cannot propose first; the third concentrates personal data in one company and raises the risk of leakage and misuse. None of them is solved unless the operating system takes charge of per-person resource management and access control.

3. Composition of aios — Four Components

Following the three conditions of an operating system, the four components of aios divide as shown below. The four components are the same regardless of deployment form (Chapter 7), and they use existing infrastructure such as cloud and AI models as parts rather than replacing it.

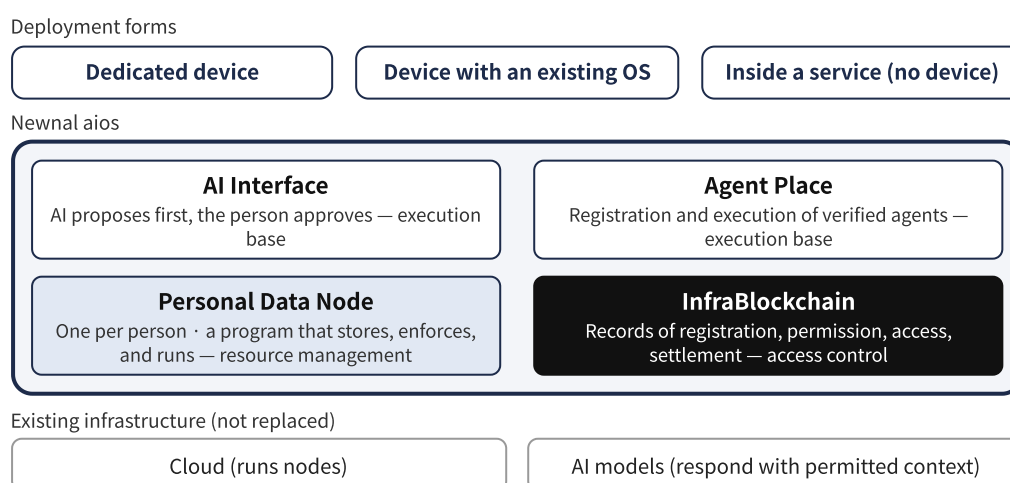


Figure 1. The four components of aios. Above are the deployment forms (Chapter 7); below is the existing infrastructure aios uses as parts.

Condition	Component	What it does
Resource management	Personal Data Node	A node running one per person stores the person's data in the person's name, and aios creates, isolates, protects, and moves nodes.
Access control	InfraBlockchain	All records of ownership, permission, and access are verified on a public ledger outside any company. Because it is a public ledger that operates without issuing cryptocurrency, companies and governments can use it, and not even Newnal can change the records.
Execution base	AI Interface	AI proposes first based on the node's data, and the person only approves.
Execution base	Agent Place	Verified agents extend services on top of node data the person has permitted.

Two of the four components need further explanation. Agents registered in **Agent Place** meet two conditions: the scope of data they read and write is registered on the public ledger, and they can prove through remote attestation that their running code is identical to the published code. Agents that meet both conditions are called "verified agents," and agents built by partner companies and by outside developers are registered under the same conditions. Usage is recorded in a per-DID ledger and settled automatically. The **AI Interface** is a cycle of proposal and approval. The AI makes a proposal using the node's data, the person approves or declines, and the approval is recorded on the public ledger as a permission record. It operates without the person issuing commands, but nothing is executed without the person's approval.

4. Managing per Person, Not per Device — The Personal Data Node

To carry out the three conditions of Chapter 1 per person, "all of one person's data and execution on that data" must form a single unit that the operating system can handle. Newnal aios sets that unit of management as the **Personal Data Node**.

A Personal Data Node is a program that runs one per person. The node encrypts and stores the person's data, enforces the scope the person has permitted, and runs services and AI agents in the Personal Data Node environment. The node runs in a confidential computing environment in the cloud, is registered on the public ledger under the person's DID, and can be opened only with the encryption key on the person's device. The person can move the node to another cloud. A person's data gathers in the node under the person's name regardless of its source: data created by devices, credentials issued by institutions, and data recorded by services with the person's permission all reside in one node.

Personal data stored in a Personal Data Node may carry the digital signature of its issuer, which makes it possible to verify both the issuer and the integrity of the data.

Services can use data the person has permitted to produce new analytical, processed, or personalized data, and return it to the person's node with their own digital signature. Data accumulated in this way can be used again by other services with the person's permission.

The node corresponds to the three conditions of an operating system as follows.

Condition	Role of the Personal Data Node
Resource management	The resource aios manages is personal data, and the unit of management is the node. aios creates nodes (one per person), isolates them (separation between nodes), protects them (encryption and confidential execution), and moves them (between clouds). aios does for nodes what existing operating systems do for files and processes.
Access control	The node enforces access control. When a service or agent requests access to the node, the person grants permission, and the node does not execute calls outside the permitted scope. Records of permission and access remain on the public ledger outside the node.
Execution base	The node runs AI agents. Agents produce personalized results in the Personal Data Node environment, using data within the scope the person has permitted, and can process the data without sending it outside.

Without the node, aios cannot exist: there would be no unit of resource to manage, no party to enforce permissions, and no base on which to run agents. Conversely, with the node, all three conditions are carried out in a single node. The node is the unit of management in aios not by design choice but because the definition requires it.

The node and the device relate as follows. The node runs in the cloud. The key that opens the node exists only on the person's device. Whether aios is installed on a device or in a service, it opens the node with the key on the person's device and runs agents. Whether the device is a dedicated device or a smartphone with an existing operating system does not change how aios works. The node keeps running even when the device is powered off, and if the device is lost, the node's data is recovered through the recovery method the person has chosen. The node's structure, key management, and security are explained in the separate document "Personal Data Node."

5. The Shift in Interface

5.1 Eras of computing defined by interface

Eras of computing have been defined by the interfaces of their operating systems. Consumers experienced each new kind of computing not through the kernel or the way resources were managed,

but through the interface.

Era	Operating system	Interface	How it works
Personal computing	macOS, Windows	Graphical interface	The person gives commands with a mouse and keyboard.
Mobile	iOS, Android	Multi-touch	The person picks an app and gives commands with a finger.
AI	aioS	AI Interface	AI proposes first based on the node's data, and the person approves.

The interfaces of the first two eras are both user interfaces: the person commands and the machine executes. The aioS interface runs in the opposite direction. AI proposes first based on the person's data, and the person only approves.

5.2 Conditions for the AI Interface

For AI to propose first, the AI must know all of the person's data, and the person must permit access. Both conditions hold only when a node exists. Only when one node stores all of the data can the AI know all of it, and only when access control runs in the node can the person grant permission. Voice assistants on existing operating systems work only with data inside the device and within their own company's services because these two conditions are absent. The AI Interface is a result made possible because aioS manages personal data per node, and it is where users experience aioS as a new operating system.

5.3 Example — How a single proposal is processed

The following shows, through a single proposal, how the three conditions and the interface actually combine. The example is a family phone for a senior household, but aioS inside a banking app or a hospital service works in the same sequence.

Step	What happens	Operating system condition
1	The device records in the person's node lifestyle data showing no movement since the previous night. A checkup result (credential) issued by the hospital is also in the same node.	Resource management — data gathers in the person's node.
2	The family care service's agent requests access to the node. The scope it can request is only what is registered on the public ledger (movement and check-in responses); the checkup result is outside that scope, so the request itself is not executed.	Access control — permission is enforced at the node.

Step	What happens	Operating system condition
3	The agent makes a judgment in the Personal Data Node environment using data the person has permitted, and proposes first: "Shall I place a check-in call today?" The data has not been sent outside the Personal Data Node environment.	Execution base — the agent runs in the node.
4	The person (or a guardian) approves on the device. The approval and the access are recorded on the public ledger, and a notification is sent.	AI Interface — AI proposes, the person approves.
5	The call connects. The connection runs over a DID-based closed network rather than phone numbers, so unregistered parties have no path to connect.	Access control — contact is also subject to permission.
6	Service usage is recorded in a per-DID ledger and fees are settled. The family care service company never stored the person's data.	Resource management — settlement is also per node.

6. Applying the Definition to aios — How It Differs from Existing Operating Systems

Condition	aios
Resource management	Manages personal data in units of Personal Data Nodes. Nodes run one per person, are registered on the public ledger under the person's DID, and open only with the person's key. On dedicated devices, aios also manages the device's hardware.
Access control	Permission from the person identified by DID. AI agents call and use only data within the permitted scope from the node, and records of permission and access remain on the public ledger (InfraBlockchain).
Execution base	Runs AI agents. Provides the AI Interface, in which AI proposes first based on the node's data, and Agent Place, where verified agents are deployed and run.

aios carries out all three conditions. It differs from existing operating systems in three respects: the resource it manages is personal data and its unit of management is the node; access control runs at the node rather than inside the device; and what it runs is AI agents rather than apps. That it is an operating system follows from the definition itself.

	Existing operating systems	aios
Resources managed	Device hardware and files on the device	Personal data (including hardware on dedicated devices)
Unit of management	Files, processes	Personal Data Node (one per person; a program that stores data and runs)
Where access control runs	Inside the device	The node, which enforces every access request from services and agents
Who grants access	The device user	The person identified by DID
Where permission records are kept	Inside the device	A public ledger outside any company
What it runs	Apps	AI agents
Interface	User interface, where the person commands	AI Interface, where AI proposes first

In existing operating systems, the user logs in to an app, the app takes the user's data, and the company that built the app decides how the data is used afterward. In aios, services and AI agents request access to the person's node and use data within the scope the person has permitted. Newnal calls this reversal of the access relationship Reverse Login.

Stated in one sentence according to the definition: aios is an operating system that runs and protects personal data in units of Personal Data Nodes, controls access to nodes based on the person's permission, and runs AI agents in the Personal Data Node environment.

7. Deployment Forms

aios is deployed in three forms. In every form, it runs and protects nodes, controls access, and runs AI agents in the same way.

Newnal aios	Dedicated device	Device with an existing OS	aios inside a service (no device)
Hardware management	Performed by aios	Performed by the existing OS	Not applicable
Personal Data Node management	Performed by aios	Performed by aios	Performed by aios

Newnal aios	Dedicated device	Device with an existing OS	aios inside a service (no device)
Access control	Performed by aios	Performed by aios	Performed by aios
AI Interface	On the device screen	Inside the app	Inside the partner's app, web, or support channel
AI agent execution	Performed by aios	Performed by aios	Performed by aios (Personal Data Node environment)
Examples	Family phone	Android TV, smartwatch	Banking app, AI contact center, hospital service

On a dedicated device, aios runs as an independent operating system that also manages the hardware directly. On a device with an existing OS, it delegates hardware management to that OS and runs on top of it as the security layer for the Personal Data Node. In a service without a device, it runs as the node runtime in the cloud and as the aios layer inside the partner's service. All three forms carry out the three conditions of an operating system.

8. Scope of Adoption — Phased Adoption

Companies do not need to adopt all four components of aios at once. They can start with the data layer and expand the scope step by step, and at every scope what they adopt is aios. The Personal Data Node and InfraBlockchain are always adopted together, because node registration and permission records reside on the public ledger. The minimum unit of adoption is therefore not the node but the aios data layer.

Scope	Components included	Benefits to the company
Data layer	Personal Data Node + InfraBlockchain	Use of data based on the person's permission, the combined reach of data spheres, verification of issuer and integrity, and accumulation and reuse of data produced by services
+ AI Interface	The above + AI Interface	Within the company's apps and channels, AI proposes first using the node's data, and the customer only approves.
+ Agent Place	The above + Agent Place	Verified external agents perform functions the company did not build, on permitted data.
Dedicated device	All of the above + hardware management	A device experience designed for aios from the start

Whatever the scope of adoption, a Personal Data Node is created for each customer. What Newnal supplies is aios, and Newnal's key metric is the number of active nodes.

9. Anticipated Objections and Answers

Without its own kernel booting the device, it is not an operating system. The definition does not make owning a kernel a condition. Android is an operating system that runs on the Linux kernel. On dedicated devices aios boots the device, kernel included; on devices with an existing OS it delegates hardware management to that OS. Either way, the three conditions are carried out the same way.

If it runs on top of an existing OS, it is an app. What separates an app from an operating system is not where it runs but its role. An app receives resource management, access control, and an execution base from the operating system; the operating system provides them. aios manages personal data in units of Personal Data Nodes, controls access to nodes, and provides an execution base to other programs (AI agents). Apps do none of these three things.

Personal data is not hardware, so it is not an operating-system resource. The resources in the definition are not limited to hardware. The file system is a core component of every operating system, and files and data are resources the operating system manages. The Personal Data Node that aios manages is a resource that combines data with execution on that data, and it takes, per person, the place that files and processes occupy in existing operating systems.

A Personal Data Node is just another name for a database or cloud storage. Databases and cloud storage store and retrieve data, but they provide no execution base to other programs, and their access control is based on the accounts of the company that owns the storage. A Personal Data Node bases access control on the person's DID and permission, does not execute calls outside the permitted scope, and runs agents in the Personal Data Node environment. The cloud on which the node runs and the storage where its data resides are parts used within aios, not substitutes for the node.

An AI agent is just another name for an app. An app takes the user's commands and works with data on its company's server. An AI agent calls data within the scope the person has permitted from the node, proposes first, and cannot use data outside the permitted scope. Because the programs being run behave and access data differently, the execution base must also be different. The execution base aios provides is an interface that handles an agent's access requests, the person's permission, and the record of that permission, and existing operating systems do not have it.

Voice assistants on existing operating systems are also AI interfaces. Voice assistants on existing operating systems carry out the person's commands within the device's data and their own company's services. They are a form of user interface in which the person commands, and their reach is the same as the reach of the resources the operating system manages. An interface that proposes first based on all of a person's data is possible only when the operating system manages personal data per node, and the aios AI Interface works on the node's data.

If it does not drive the device, it is not an operating system. The three conditions of the definition do not include driving hardware. There is precedent. ROS (Robot Operating System) runs on Linux and handles neither the kernel nor the hardware directly, yet because it abstracts robot resources and provides communication between components and an execution base, the robotics industry worldwide classifies it as an operating system. Even inside a service with no device, aios runs and protects Personal Data Nodes, controls access to nodes, and runs AI agents. A device is the case in which aios also manages hardware, not a condition for being an operating system.

"Operating system" is a marketing term. The test is the definition, not the label. The table in Chapter 6 shows, item by item, whether each of the three conditions is carried out. If someone argues that any one of the three is not carried out, the discussion can proceed on that item.

Conclusion

The definition of an operating system requires three conditions, resource management, access control, and an execution base, and driving hardware is not among them. Existing operating systems carry out the three conditions for the device's files and processes and for apps. Newnal aios carries out the same three conditions per person. It runs and protects personal data in units of Personal Data Nodes, controls access to nodes based on the person's permission, and runs AI agents in the Personal Data Node environment through an interface in which AI proposes first. It works with or without a device, and on top of an existing operating system or without one, and whatever form it is adopted in, a Personal Data Node is created for each customer. In aios, the source and integrity of data carrying an issuer's digital signature can be verified, and new data produced by services also accumulates in the person's node and can be reused by other services.

References Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 4th ed., Pearson, 2015, Chapter 1. / Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, 10th ed., Wiley, 2018, Chapter 1.