



OCTERYX OS / FOUNDATION GUIDE 01

Autonomy needs a memory.

A machine can perceive the present. Dependable autonomy requires a governed account of what happened before, what matters now and what may happen next.

FOUNDATION EDITION 0.1 / JULY 2026

 MARK NICOLL - FOUNDER



Contents

A field guide to the governed context infrastructure beneath autonomous systems.

- | | |
|----|---|
| 01 | The missing layer is not another sensor. |
| 02 | Data is not continuity. |
| 03 | Context operating system for autonomous machines. |
| 04 | Hardware changes. Context should remain. |
| 05 | Memory is only useful when it is governed. |
| 06 | From signal to context, action and evidence. |
| 07 | The right context is not the most context. |
| 08 | Care makes the context problem impossible to ignore. |
| 09 | Dependable autonomy begins with explicit boundaries. |
| 10 | The next actor should inherit more than an empty session. |
| 11 | Context readiness canvas. |
| 12 | A category thesis is not a production claim. |
| 13 | Octeryx OS is building the context beneath autonomy. |

[Read the canonical web edition and use the interactive working canvas.](#)

The missing layer is not another sensor.

Autonomous systems are becoming better at seeing the present. They remain surprisingly poor at carrying forward what the present means.

A camera can detect a person. A model can classify an event. A robot can map a room. A workflow can record that a task happened. None of those facts, by itself, tells an autonomous system whether it has identified the right person, whether the observation is still current, whether the source should be trusted, whether an action is permitted or how the decision should later be reconstructed.

The autonomy stack has concentrated enormous effort on perception, planning and action. Those capabilities are essential. But physical environments are not blank spaces that reset between model calls. They contain people, history, permissions, responsibilities, routines, uncertainty and consequences.

Dependable autonomy therefore needs a layer that persists across devices, models, applications and handovers. It must turn isolated signals into governed context: a usable account of who and what is involved, what has changed, how confident the system should be, which policy applies, who must review and what evidence should remain.

Perception describes a moment. Context explains what the moment means.

This guide explains that missing layer. It is written for the organisations building autonomous systems, the operators expected to trust them and the investors, researchers and press trying to understand where the next infrastructure category may emerge.

Data is not continuity.

An environment may be saturated with information while every actor still operates from a different version of the world.

Robots keep local maps and mission state. Sensors emit readings. Models produce scores. Applications store workflow records. Humans carry knowledge in memory, conversation and notes. Each source can be individually useful while the operating environment remains fragmented.

Fragmentation becomes visible at the interfaces. A device detects movement but cannot establish who moved. A model flags a change but cannot see that the underlying fact has expired. A robot receives an instruction but not the permission boundary around it. A new application asks the same identity and history questions because the previous answers were trapped elsewhere.

The problem is not simply that data lives in several systems. The deeper problem is that meaning, authority and time travel poorly between them.

Question	Isolated signal	Governed context
Who?	A detected entity or local identifier	A resolved identity with ambiguity and evidence visible
When?	A timestamp	Freshness, validity and revalidation state
May it act?	A capability is available	Policy, purpose, authority and human ownership are applied
Why?	A model or rule produced an output	The source, context, policy and decision path can be reconstructed

Context operating system for autonomous machines.

A governed runtime that helps robots, sensors, software systems and human operators act from shared context across physical environments.

The category sits between the systems that perceive or record the world and the actors that decide or act within it. Its purpose is not to replace robot control, safety systems, domain applications or human judgement. Its purpose is to make the relevant context persistent, governed and portable across them.

A context operating system receives observations and requests, associates them with the right identities, materialises the context appropriate to the actor and purpose, applies policy and confidence, coordinates the permitted response and preserves the evidence required to understand what happened.

Octeryx OS is being designed around that category. The public architecture separates the context runtime from the underlying identity, context-materialisation and provenance infrastructure, while keeping the product understandable to organisations that do not need an internal systems lecture.

Not	A robot control stack	Navigation, motion control and hardware safety remain with the machine and its OEM systems.
Not	A detached workflow database	Applications can present and coordinate work without becoming the owner of canonical context.
Is	A governed context runtime	Identity, memory, policy, coordination and evidence persist across actors and interfaces.

Hardware changes. Context should remain.

The most valuable thing an autonomous environment accumulates may not be a device or model. It may be the governed account of how the environment works.

Machines are replaced. Models are upgraded. Operators change shifts. Applications are procured and retired. Environments expand. If identity, history, permissions, routines and evidence live only inside each component, every change forces the operation to rebuild part of its understanding.

A durable context layer separates what the environment knows from the particular interface currently consuming it. A new robot should not need to become the first historian of the site. A new model should not quietly erase the basis of earlier decisions. A new operator should not inherit only whatever happened to be written in the previous system.

This is why operational context can become a compounding asset. Each governed observation, resolution, policy decision and handover can make the next permitted interaction more informed, while provenance and freshness keep that memory from becoming unchallengeable folklore.

01	New robot	Identity, environment and operating history remain available through a new actor contract.
02	New model	The model changes without becoming the sole owner of memory or decision evidence.
03	New interface	The workflow changes while governed context remains beneath the presentation layer.
04	New setting	Relevant context moves under explicit policy, identity and handover boundaries.

Memory is only useful when it is governed.

An autonomy context layer must remember more than events. It must preserve the conditions under which those events may be trusted and used.

01	Identity	Resolve people, machines, objects, places and systems while keeping ambiguity visible.
02	Memory	Preserve longitudinal context instead of resetting at each device, session or application.
03	Freshness	Show when a fact, permission or inference may have expired or require revalidation.
04	Policy	Shape what each actor may see or do for a stated purpose and operating boundary.
05	Coordination	Route the permitted instruction, prompt, handover or escalation to the responsible actor.
06	Evidence	Retain the lineage from source to context, policy and outcome for later reconstruction.

These requirements reinforce one another. Identity without policy can disclose the right person to the wrong actor. Memory without freshness can preserve a dangerous obsolete fact. Coordination without evidence can make action impossible to explain. Evidence without clear truth ownership can create another shadow record.

The design question is therefore not whether the system can remember. It is whether memory, uncertainty, policy and accountability remain attached as context moves.

From signal to context, action and evidence.

The public Octeryx architecture gives each responsibility an explicit owner rather than compressing the whole problem into a single black box.

01	Physical or care environment	Humans, robots, sensors, applications and systems emit observations or request context.
02	Octeryx OS	Normalises the event, applies the actor contract and coordinates the governed route.
03	Makemake Onboard	Resolves identity and makes ambiguity explicit before unsafe identity use.
04	ECP Onboard	Compiles relevant context and evaluates confidence, validity, authority and policy.
05	CongDB Onboard	Commits context, truth-lane, decay, provenance and evidence records to PostgreSQL.

The path is intentionally directional. Environment actors should not write around the governed runtime. Context should not be materialised without policy. Identity-bearing events should not become canonical while the subject remains unresolved. Presentation state should not quietly outrank the provenance-bearing graph.

The target architecture is more important here than the internal names. It makes a commercial promise legible: Octeryx OS is intended to preserve continuity and accountability across a heterogeneous operating environment without pretending to own every device or application.

The right context is not the most context.

A shared context layer should not become a universal data dump. Each actor needs a permitted, purpose-shaped slice.

A carer, family member, robot, maintenance provider and autonomous agent may all participate in the same environment while requiring very different information. Sharing the entire record is unnecessary and often unsafe. Sharing too little makes coordination brittle.

The useful unit is therefore a context slice: the minimum relevant combination of identity, history, current state, policy and evidence required for a specific actor and purpose. The slice should carry its own boundaries, including freshness, confidence, expiry and prohibited use.

Human operator	Decision context	What changed, why it matters, what requires review and which evidence supports it.
Robot	Task context	The identity-safe, purpose-limited context required to perform or decline a task.
Family	Relationship context	The permitted update for the right person, purpose and disclosure boundary.
Model or agent	Inference context	Source quality, freshness, policy and uncertainty attached to the selected inputs.

Shared context does not mean shared access. It means coordinated action from one governed account of the environment.

Care makes the context problem impossible to ignore.

Identity, continuity, policy, freshness and human responsibility are not optional refinements when context can affect a person receiving care.

Care environments contain the full context challenge in concentrated form. Residents move between homes, care homes and hospitals. Families, carers, clinicians, devices, voice systems, applications and future robots see different fragments. Permissions and responsibilities vary by actor, setting, purpose and time.

A medication exception, mobility change, family concern, handover note or sensor signal can be individually accurate while still being dangerous when separated from identity, timing, authority or the rest of the resident's context. A safe response often depends less on the volume of data than on whether the right facts are current, permitted and connected.

That is why care is the primary mission for Octeryx OS. It exposes the infrastructure requirements early. Kasa is the care profile and care-facing surface inside Octeryx. Film is one isolated adjacent profile, not permission for generic sector expansion.

Dependable autonomy begins with explicit boundaries.

The context layer becomes credible when every actor can see what it owns, what it consumes and what it must never quietly become.

Humans	Care and operating decisions	
Robots	Perception and physical capability	
Applications	Presentation and workflow interaction	
Octeryx OS	Governed context orchestration	
Makemake Onboard	Identity and entity resolution	
ECP Onboard	Context compilation and policy	
CongDB Onboard	PostgreSQL context, truth and evidence	

A robot may perceive and act without owning resident memory. An application may coordinate a workflow without becoming the system of record. A model may contribute an inference without turning probability into unqualified truth. A human may retain decision authority without being forced to reconstruct the entire machine path unaided.

Octeryx OS is intended to sit above onboard control, navigation and hardware safety systems. It does not replace them. It also does not claim autonomous clinical decision-making, unsupervised robot operation or replacement of carers, clinicians or human operators.

The next actor should inherit more than an empty session.

Context infrastructure becomes more valuable when each governed interaction improves the environment's ability to orient the next permitted actor.

Most technology deployments restart learning at the component boundary. A device knows its own history. An application knows its own workflow. A model sees the inputs supplied to one call. A new actor enters with limited access to what the environment has already established.

A context operating system changes the unit of accumulation. The environment can preserve resolved identity, source trust, policy outcomes, changes over time and evidence independently of the device or interface that first produced them. Each new actor can begin from a governed context slice rather than from zero.

The compounding effect is not automatic. Poor observations, stale facts and weak provenance should not become stronger merely because they were stored. The system must preserve uncertainty, decay and review requirements alongside memory. Good context infrastructure compounds accountable understanding, not unchecked certainty.

01	Observe	An actor contributes a signal with source and purpose attached.
02	Resolve	Identity, confidence, validity and policy shape what the signal can mean.
03	Coordinate	The permitted context or action reaches the responsible actor.
04	Evidence	The outcome and lineage improve the next governed interaction.

Context readiness canvas.

Use this before an autonomous pilot, integration or investment decision to find the first place where meaning, authority or evidence will disappear.

PRINTABLE CONTEXT READINESS CANVAS

Record the status, owner and evidence for each area. The first unowned boundary is the most important architecture gap.

AREA	CONTEXT QUESTION	STATUS / OWNER / EVIDENCE
Identity	Can every person, device, place and actor be resolved, with ambiguity handled?	
Continuity	Does useful context persist when the device, model, operator or interface changes?	
Source trust	Is authority and reliability explicit for every contributing source?	
Freshness	Can the system show when context expires, decays or needs revalidation?	
Policy	Is access and action shaped by actor, purpose, setting and authority?	
Human review	Are decision ownership, escalation and fallback pathways explicit?	
Coordination	Can a permitted handover or action reach the responsible actor without hidden bypasses?	
Evidence	Can the path from source to context, policy and outcome be reconstructed?	

A low score is not a reason to abandon an autonomous use case. It is a reason to place the missing context responsibility inside the design before the pilot creates its own accidental answer.

The first unowned area is usually more important than the total. If identity is ambiguous, policy is implicit or evidence disappears at a boundary, additional capability can make the system less governable rather than more useful.

A category thesis is not a production claim.

Octeryx OS is in active R&D. The current public stage is accepted product architecture and deterministic repository proof.

Currently evidenced	Not currently claimed
- Accepted modular product doctrine, architecture and implementation plan - Historical typed Kasa contracts and compatibility boundary proofs - Deterministic fixtures, replay and repository validation - Target Onboard, PostgreSQL, Kasa and Film boundaries - Current claim, risk, data and jurisdiction controls	- Live production deployment - Clinical validation or regulatory approval - Certified operation or production readiness - Universal robot compatibility - Unsupervised autonomous decision-making or robot operation - Replacement of human operators, care teams, OEM software or safety systems

The distinction matters because autonomy infrastructure will be judged partly by how accurately it describes itself. Architecture, typed boundaries, fixtures and deterministic replay are meaningful evidence. They are not the same as live operation, external validation, certification or production safety.

This guide explains the intended category and target architecture. It should be read alongside the canonical [Octeryx OS facts](#) and [proof boundary](#), which are maintained as the public source of current claims and non-claims.

Octeryx OS is building the context beneath autonomy.

A context operating system for safe, explainable physical-world autonomy, with Kasa as the primary care profile and Film as one isolated adjacent profile.

Category	Context operating system for safe, explainable physical-world autonomy
Care framing	Resident-context operating layer for autonomous care
Core thesis	The operational context is the durable asset
Current stage	Active R&D / Accepted architecture and deterministic repository proof
Infrastructure relationship	One Octeryx product with Makemake Onboard, ECP Onboard and CongDB Onboard concepts; Kasa and Film are isolated profiles

This guide is educational material, not a clinical, safety, engineering, procurement, regulatory, legal or investment recommendation. Product and maturity statements should be read with the current Octeryx OS proof boundary.



OCTERYX OS

Autonomy Needs a Memory

Octeryx OS is building a context operating system for safe, explainable physical-world autonomy, with Kasa as the primary care profile and Film as one isolated adjacent profile.

octeryx.com



MARK NICOLL - FOUNDER

Educational material only. This publication is not a clinical, safety, engineering, procurement, regulatory, legal or investment recommendation. Product and maturity statements should be read with the current Octeryx OS proof boundary.