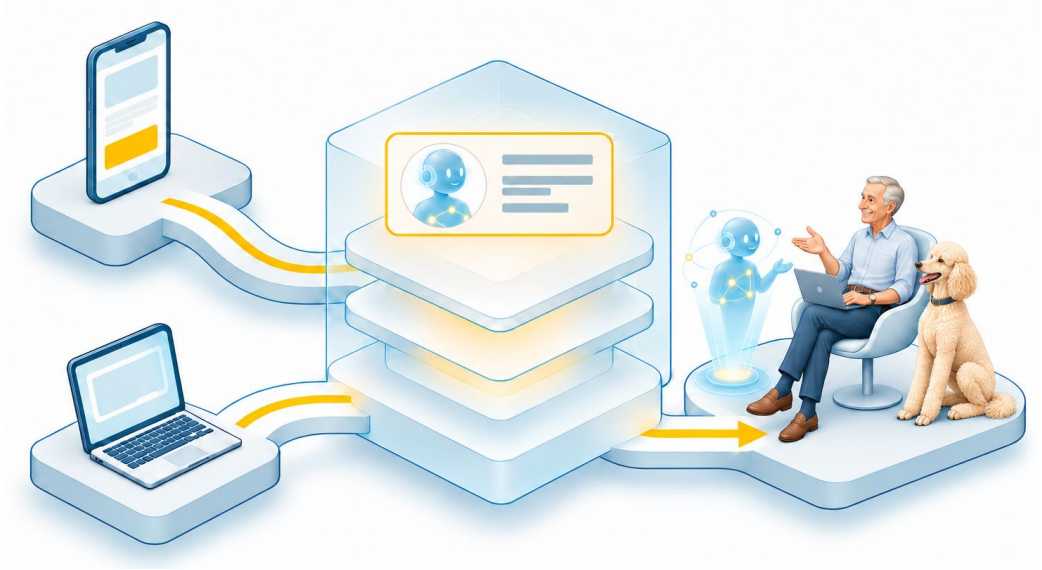
ARTEMIS PETCONNECT INSIGHTS

Everything Begins with Understanding

PAPER III / WHERE WE ARE HEADED

Teaching Applications to Be Understood



Artemis Petconnect

Creator of askKynos™

November 2026 • White Paper III

© 2026 Artemis Petconnect, Inc. Licensed under CC BY-ND 4.0.

Why capable agents need applications that can explain themselves

By two of its co-founders, the CEO & President and the CFO & CTO, Artemis Petconnect, Inc. · November 2026

Executive summary

In early 2026, our CEO imagined opening askKynos™ and simply asking Kynos, our animated companion, for help: “Can you find me a sitter for next Saturday for Wong? I need to be away for a couple of days, and I want someone who is a good fit.” Less than a year later, that same request might begin somewhere else entirely, inside whichever AI assistant a pet parent already uses. The voice may change. What has to stay the same is everything that makes the answer trustworthy.

That is the argument of this paper. Most of the conversation about AI focuses on making agents more capable. We propose a complementary priority: make applications easier for agents to understand. An agent acting on someone’s behalf needs to know what an application can do, what it requires, what rules apply, and what has actually happened. Capable agents and well-described applications are not substitutes. Together, they are a far better basis for useful, controlled action.

We describe a practical framework for doing that, six things an understandable application makes explicit, and a readiness scale for judging progress. We use our own plans for askKynos as the case: outside assistants may become an additional way in, while askKynos stays responsible for each pet’s story, Smart Match, caregiver approval, booking and payment. We have begun building the internal foundation for that work, for our own use first: an internal MCP that will let the four of us ask our own AI questions like “What’s holding back sitter signups in Orlando?” and find the bottlenecks in our growth.

> **Scope**
>
> Standards may change. Customer outcomes and accountable application boundaries should endure. This paper sets out a design direction, not a description of features available today: no outside AI assistant can connect to askKynos now, and no partnership or integration exists with any assistant named here. The research cited describes the industry, not askKynos. Figures are conceptual, not a disclosure of our architecture.

1 The original vision, and what changed

The original idea was never “a list of nearby sitters.” It was a recommendation that fit Wong: an older standard poodle who is happiest receiving care in his own home, with as little disruption to his routine as possible. That outcome has not changed.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

Neither have the responsibilities that make it possible. askKynos must keep each pet's context, run Smart Match, offer contextual guidance through its Vet Assistant, approve caregivers through an accredited background-check provider, take payment through Stripe, record real acceptance and confirmation, and support updates, photos, messaging and notifications while care is under way. These are product responsibilities, not tasks to hand to an outside assistant.

What changed during 2026 was the expected front door. Instead of building our own conversational agent to take booking requests inside askKynos, we now expect a pet parent might begin with ChatGPT, Claude, Grok or an assistant that does not exist yet. Kynos is not going away. Built with Rive and given its own voice, Kynos stays at the heart of the app: greeting people by name, asking about their pet, celebrating a confirmed booking, and growing with new moments over time. What changes is that Kynos does not become the agent you ask to arrange care. That job belongs to whichever general-purpose assistant a person already uses, and since no one knows which will win, we plan to build interfaces for more than one.

Different entry points, the same trusted outcome

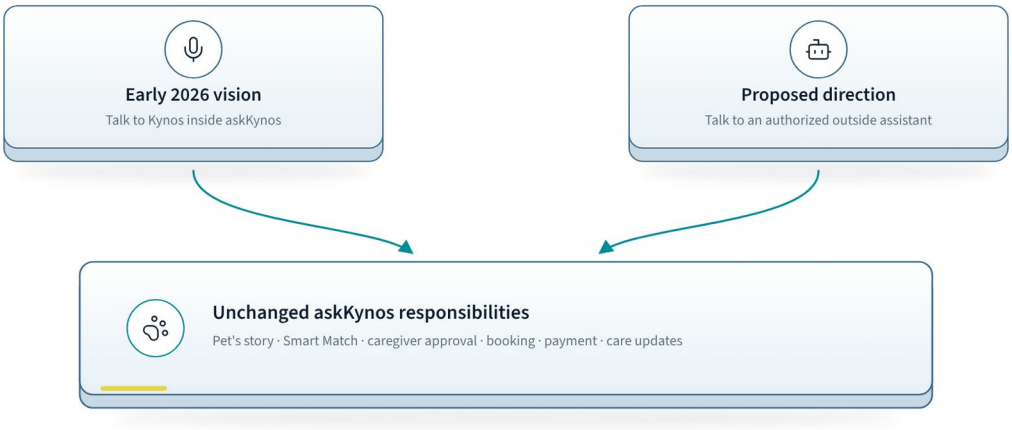


Figure 1. Different entry points, the same trusted outcome. The early-2026 vision (talk to Kynos in askKynos) and the proposed direction (talk to an authorized outside assistant) both rely on the same askKynos responsibilities.

In early 2026, the imagined voice was Kynos. By late 2026, it might belong to any of several assistants, or to one not yet invented. The request is still recognizable: find a caregiver who fits Wong, preferably in Wong's own home, with as little disruption as possible.

> The enduring requirement is understanding Wong, not owning the microphone.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

2 The next interface problem is not another screen

Business applications were designed for people using screens. APIs later opened many application functions to other software. AI agents add a new challenge: the application must be intelligible to software trying to fulfill a person's request, often across different products.

An agent asked to arrange care for a senior pet may grasp the goal and still not know which services are offered, whether there is a verified pet profile, what it may share, whether availability is current, or whether a booking is merely prepared rather than confirmed. Fluent language cannot supply those facts.

When an application is opaque, agents fall back on guesses, screen-scraping or loosely read documentation. That sometimes works. In high-consequence workflows it is brittle. A better design lets the application declare useful facts about itself while keeping responsibility for the result.

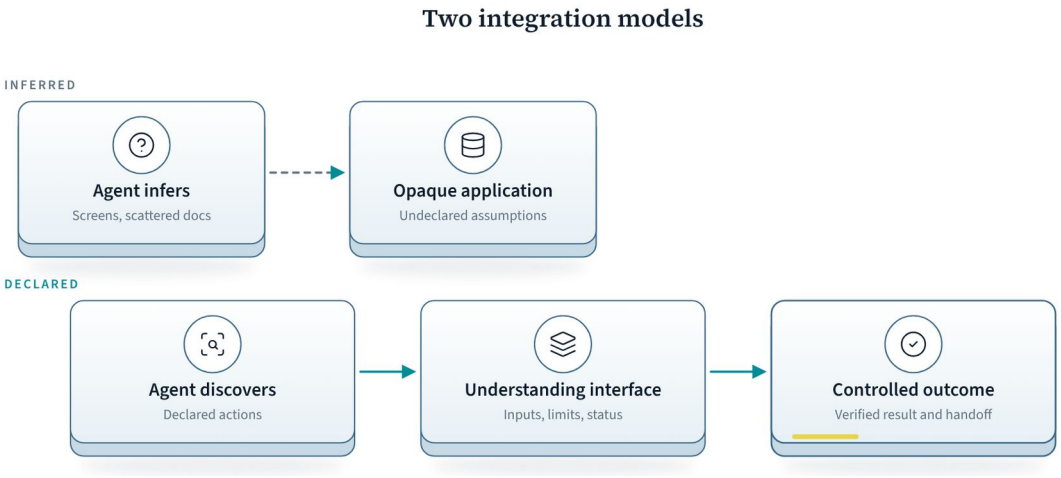


Figure 2. Two integration models. An agent inferring an opaque application's behavior, contrasted with an agent discovering declared capabilities and receiving a controlled outcome.

3 What it means for an application to be understood

Understanding is not the same as access. An agent may be permitted to call a function without knowing when calling it is appropriate. It may know the input format without knowing what the action commits the customer to. It may receive a success-shaped reply without knowing whether a transaction is pending, accepted, paid or final.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

We propose that an understandable application make six things explicit. Engineers will recognize the idea: it is, in effect, an interface control document written for agents.

Category	The question it answers	An askKynos example
Capability	What can actually be done?	Search for caregivers for a specific service
Preconditions	What must be true first?	A verified pet profile exists
Context	What information is relevant and permitted here?	Only the pet details this caregiver needs for this stay
Constraints	What will the application not do or promise?	It never confirms availability it does not have
Authority	Who may approve, revoke or execute?	The parent chooses the caregiver; the caregiver accepts
State	What happened, and what is unfinished?	Request submitted is not caregiver accepted, which is not paid, which is not confirmed

Here is how the six categories work for one real action.

> **A capability card: “Find caregivers for house sitting”**
>
> **Capability:** return caregivers in range who fit this pet and this service, ranked by fit.
>
> **Preconditions:** the parent is signed in, and the pet has a verified profile.
>
> **Context:** the pet’s care needs for this service, the dates and the general area. Never the full medical history, and never the home address at the search stage.
>
> **Constraints:** it never invents availability, never chooses for the parent, and never says a caregiver has accepted before they have.
>
> **Authority:** the parent chooses; the caregiver accepts or declines; payment happens only inside askKynos.
>
> **State:** search returned, then request submitted, then caregiver accepted, then paid, then confirmed. Each one is reported separately.

An assistant that reads this card knows what it can ask for, what it must not assume, and how to tell the person truthfully where things stand.

Together, these form an application understanding layer. It is a design lens, not a new standard and not necessarily a separate piece of software. Parts of it may be expressed through MCP, existing APIs, authorization services or handoffs to the app’s own screens.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

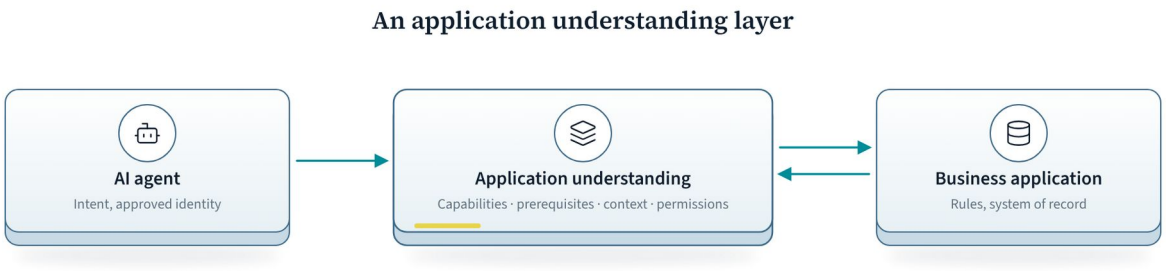


Figure 3. An application understanding layer. The agent reaches a controlled business system through explicit capabilities, prerequisites, context and permissions; the system of record stays in charge.

4 The agent industry in 2026: capable, but fallible

A credible forecast starts with evidence. Stanford's AI Index 2026 reports that agents completing computer tasks on the OSWorld benchmark reached 66.3% accuracy, up from about 5% a year earlier, against a human baseline of 72.35% [1]. That is remarkable progress, and it is still below people doing the same tasks.

Harder, more realistic tests show the gap more clearly. The ComplexMCP benchmark, presented at ICML 2026, gives agents more than 300 tools across seven stateful environments, from office suites to financial systems. Even the strongest models fail to exceed a 60% success rate, against 90% for humans. The authors name the failure modes plainly: tool-retrieval saturation, over-confidence that skips checks, and "strategic defeatism" instead of recovery [2]. A separate benchmark of real-world MCP tasks, MCPAgentBench, reports significant performance differences on complex, multi-step tool use [3].

The lesson is not that agents cannot act. It is that better models alone will not prevent wrong business actions. Applications should declare their capabilities and limits, enforce permissions and transaction state, return outcomes that can be verified, and support recovery when something fails.

5 MCP: a useful protocol, not a complete business design

The Model Context Protocol (MCP) gives AI assistants a standard way to connect to outside applications, which can offer tools, data and prompts through a common interface [4]. The July 2026 revision made the core protocol stateless, strengthened authorization, and formalized extensions, including Tasks for long-running work and MCP Apps [5]. Tasks fits a booking well: a request is submitted now, and a caregiver accepts later.

Its value is simple: applications no longer have to rely on every agent reverse-engineering a human interface. But MCP is not a distribution channel, an endorsement, or a guarantee that any assistant will surface a particular product. It does not supply an application's business rules, verify that a response is true, or decide which person must approve a transaction. Support also varies by assistant, so each one must be tested. The useful question for

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

an architect is not “Can we add an MCP server?” It is “Which capabilities can we expose safely, which context is appropriate, what must the application decide, and how does a user know an action is complete?”

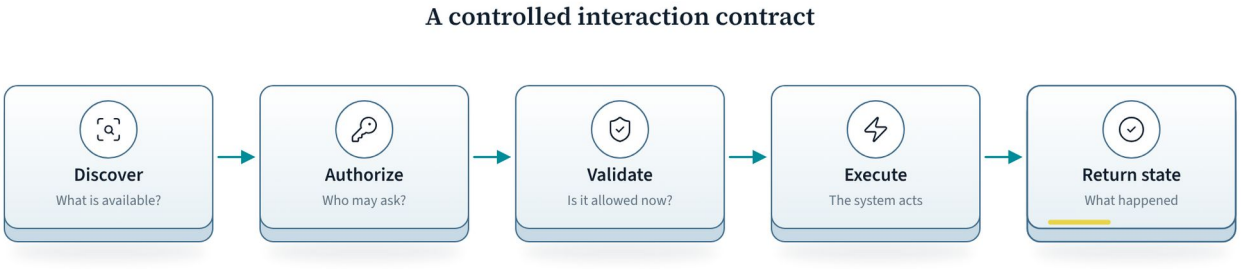


Figure 4. A controlled interaction contract. Discovery, authorization, validation, execution and returned state are distinct stages.

6 Design for many assistants, not one winner

No one can responsibly predict which assistant or protocol will lead in 2027. People will prefer different assistants, and each will have different permissions, safety policies and integration support. So we should not build product logic separately into three or five assistant-specific experiences.

There is a simpler reason as well: we would rather go with the tide than race against it. The largest AI companies are spending billions to make their assistants better at conversation. A four-person company cannot outrun that tide, and does not need to. Every improvement they make to understanding a person’s request becomes available to askKynos the day we connect. So we put our own effort where no assistant can stand in for us: understanding each pet, matching on fit, and keeping the pet’s story going between bookings. As we wrote in Paper I, we didn’t add AI to a pet-care marketplace. We built a pet-care marketplace around what AI makes possible, and that includes the assistants others are building.

Our proposed approach is one trusted capability contract: machine-readable descriptions of service discovery, pet-context permissions, Smart Match requests, handoff requirements, booking states, care updates and safe error responses. An adapter, MCP where appropriate, would expose a carefully limited part of it. Other adapters could follow if standards and demand require them. One contract does not mean zero integration work: each assistant still brings its own sign-in, consent, presentation, testing and support.

This separation protects what makes askKynos different. Outside assistants may interpret intent, collect authorized inputs and present results. askKynos stays authoritative for each pet’s context, matching, caregiver eligibility and screening, payment, confirmed booking state, messaging, care continuity and notifications. No assistant should be able to infer that a booking is confirmed merely because a tool call succeeded.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

> No assistant should be able to infer that a booking is confirmed merely because a tool call succeeded.



Figure 5. One trusted capability contract, many potential assistants. Customer intent passes through one contract (context, permissions, matching, transaction state) to askKynos, which keeps the authority. Illustrative only; no assistant integration exists today.

7 askKynos as a test case

askKynos is a working pet-care platform built on the belief that every pet deserves to be understood. That makes it a useful place to explore agent access: an outside assistant could understand a parent's intent, but the pet's history and the marketplace's decisions should stay with the application responsible for them.

Our plan for 2027 describes two channels, not a replacement. Inside askKynos, Kynos remains the companion who greets, remembers and celebrates. Outside it, assistants might introduce pet parents or sitters, explain the services, prepare permitted search inputs and guide people into the app's existing screens. An assistant may start a permitted inquiry. It should never invent matches, choose a caregiver without the person, take payment outside askKynos, or claim a request is booked when it is not.

The approach is deliberately modest: the smallest trustworthy connection to what already works, not a second booking engine or a rewrite of the customer experience.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

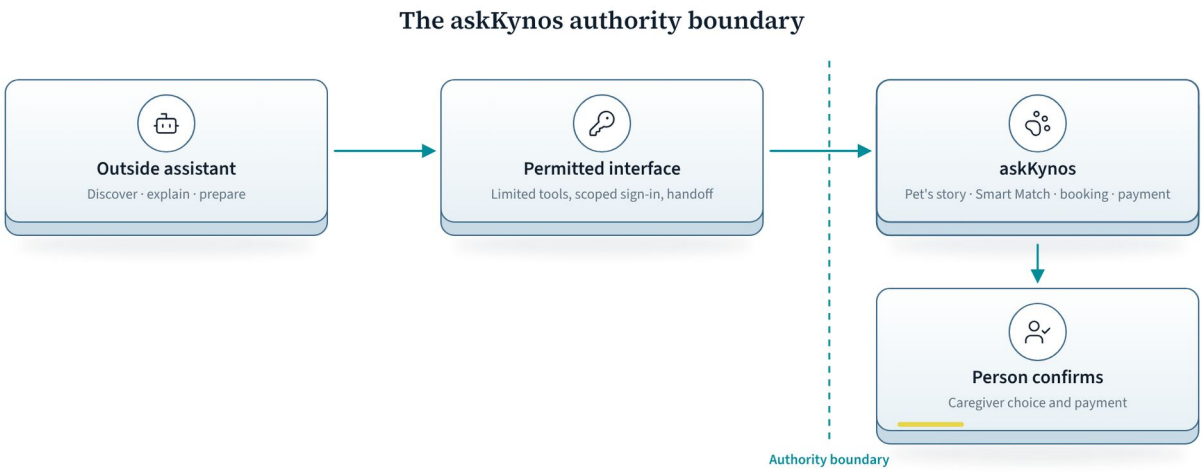


Figure 6. The askKynos authority boundary. Outside assistants discover, explain and prepare; askKynos keeps decisions, records and transactions; the person confirms the caregiver and the payment.

Three journeys

Wong's family finds care at home. Wong's family asks their assistant to find house sitting for a weekend. A future integration might recognize askKynos as relevant, explain that care is matched to the individual pet, and send the family to Wong's verified profile. If permitted search is available, askKynos, not the assistant, returns suitable caregivers, ranked by fit. The family chooses, books and pays inside askKynos. If no suitable caregiver is available, that is the answer; an assistant must never manufacture availability.

A sitter finds a better marketplace. An experienced sitter asks an assistant how to find more clients without giving up the ones they have. The assistant can explain askKynos's current terms (sitters keep their own clients and can work on other platforms) and send the sitter to the existing sign-up. Background checks, approval and payout setup stay with askKynos. A referral is not an approved sitter, and an approved sitter is not a guaranteed booking. Because sitters are our most important need, this journey may matter most in the near term, and much of it starts with clear public pages that assistants can read accurately today (Section 9).

A returning family picks up where they left off. Months after a good stay, a parent asks whether the same caregiver is available again. After sign-in and consent, a permitted interface could recall the earlier care relationship, ask what has changed, and help prepare a new request. The pet's story and the booking record stay with askKynos, whichever assistant the parent uses. This is where Paper II's compounding understanding meets this paper's intelligible application.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

8 The path from here: small steps, each one proven

We cannot forecast the agent industry years ahead. Even a three-month prediction can age quickly as models, protocols and distribution change. So we propose a testable direction, taken in stages, with each stage gated on the one before it and on the core marketplace working well.

Stage	What it tests	Gate before moving on
Now: internal foundation	An internal MCP for the four of us, starting with questions about our own growth, such as why sitters in one city aren't signing up	Reliable for daily internal use
First: read-only discovery	Outside assistants can learn what askKynos offers, using public information only	Descriptions are accurate and current
Next: guided handoffs	Pet parents and sitters are sent into the app's existing sign-up and profile screens	Handoffs keep the person's intent without exposing private data
Later: authorized search	Permitted, signed-in requests for suitable caregivers and availability	Consent, state and error recovery verified end to end

Throughout, a parent still chooses, confirms and pays inside askKynos, and a caregiver keeps the normal right to accept or decline. We expect the early stages in the first half of 2027. They are planning targets, not announced releases.

Whether assistants come to dominate discovery, share it with apps, or fragment across providers, the same things matter: accurate declarations, relevant context, limited authority, verifiable state and safe handoffs. Our rule is to build against demonstrated interfaces, not declared winners: test with at least two quite different assistants where feasible, and expand to any action that changes something only after explicit user authorization and end-to-end confirmation are independently verified. Getting listed in an assistant's directory is a separate effort with its own review process, and we plan for it separately from engineering.

The measures that matter are verified pet profiles, approved sitters, reliable handoffs, completed bookings, repeat use, consent failures, tool errors and abandoned journeys. Exposing an MCP endpoint, or appearing in an assistant's answer, is not commercial evidence.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

9 Agent Readiness Levels

A simple scale helps teams see the work in stages. We call it Agent Readiness Levels, in the spirit of the readiness levels engineers already use for new technology.

- **Level 1, human interface.** People navigate screens.
- **Level 2, documented interfaces.** Machines can call defined functions.
- **Level 3, discoverable capabilities.** Selected capabilities are described in machine-readable form, potentially through MCP.
- **Level 4, governed actions.** Identity, permissions, validation, state and monitoring are robust.
- **Level 5, contextual continuity.** Bounded context and continuity carry across interactions.

The first step does not even require an integration. In October 2026, a week after we rewrote our websites, we asked Google's AI a neutral question about askKynos and supplied nothing else. It chose the comparisons itself and described our approach correctly: care matched to the individual pet, a pet story that carries forward, and rankings that are never for sale. That is not an endorsement, and it says nothing about bookings. It does show that clear public pages already make a company understandable to AI, before any agent connects to anything.

Higher levels do not make lower ones obsolete, and not every application needs Level 5; a public information page and a payment system carry different risks. Today, askKynos is at Level 1: it was built for people using screens, and nothing in it is agent-ready yet. Our internal MCP is the first step toward Level 3, for our own use before anyone else's.

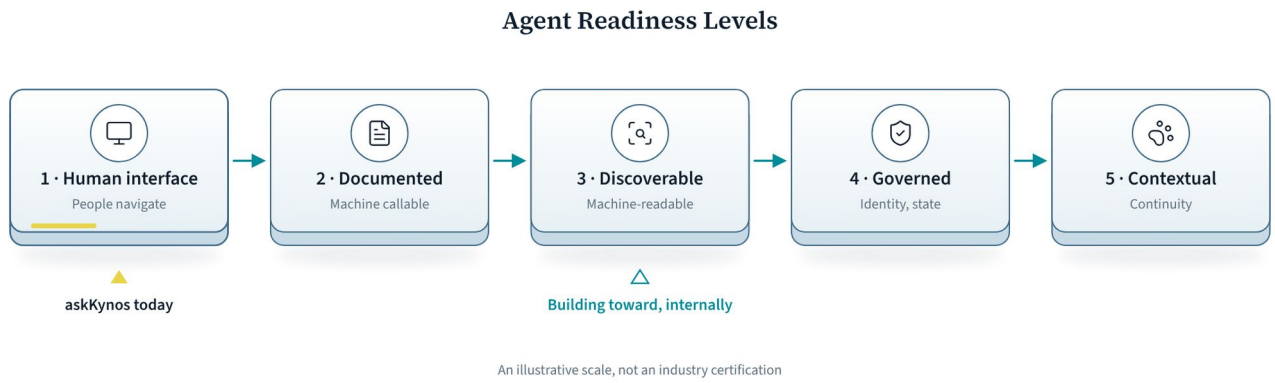


Figure 7. Agent Readiness Levels 1 to 5. An illustrative scale, not an industry certification.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

10 What responsible adoption requires

Truthful descriptions. Published capabilities must match current features, regions, rules and requirements. Being discoverable is not the same as being recommended, and both are measured separately from engineering readiness.

Consent, privacy and least privilege. Pet histories, customer details, caregiver identities and payment information need purposeful limits. An outside assistant should receive only what the authorized task requires, and an exposed capability is not blanket permission to use it. MCP's authorization framework covers the connection itself; business-level consent must still be enforced by the product [4]. Before any pilot touches real customer data, our own privacy terms and consent flows will be updated to cover it.

Defense against hidden instructions. Much of what an agent would read in askKynos is written by people: pet notes, sitter profiles, messages. Someone could hide instructions aimed at an agent inside that text. The NSA's guidance on MCP security highlights exactly this class of risk, warning about trust boundaries and implicit trust relationships in agent systems [6]. User-written text must be treated as information, never as instructions.

Integrity of actions and state. Interfaces should distinguish a preview, a submitted request, an accepted service, payment and a confirmed booking. Repeat-safe operations, durable identifiers, audit logs, explicit confirmation and clear failures matter more when natural-language requests can be repeated or interrupted.

Independent evaluation. A pilot should measure more than whether an agent called a tool: qualified profiles, approved sitters, completed bookings, repeat use, failed handoffs, authorization errors, stale availability and support incidents. Optimism is not evidence.

11 What remains unproven

A technically successful integration may attract little demand, and every new surface adds security, privacy, monitoring and operating cost. If an advantage emerges, it will come from careful implementation, disciplined experiments and evidence from real journeys, not from owning an open idea.

A protocol-independent contract is also an investment in options. If MCP stays the main connector, we extend the adapter. If other standards emerge, we change the edge without rewriting the pet's story, Smart Match or the care workflow. That flexibility is a goal to prove, not a promise.

12 Conclusion: understanding is a shared responsibility

The next generation of useful AI will need more than capable models. It will need applications that clearly say what they know, what they can do, what they need, what they will not allow, and what has actually happened.

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

MCP is one practical way to say it. The harder work is architectural and organizational: keeping trusted context, keeping business authority, enforcing permissions, and giving people a clear path from intent to a verified result.

This extends the belief behind all three papers: everything begins with understanding. In Paper I, that meant understanding complex work and keeping people accountable. In Paper II, it meant understanding an individual pet over time. Here, it means helping intelligent systems understand the applications they use, without giving up the human decisions those applications exist to protect.

The early-2026 vision did not expire; it matured. The voice may change, but the promise remains: know Wong, find the right care, complete it responsibly, and keep what was learned. We cannot choose which assistant wins. We can make askKynos understandable, trustworthy and ready for more than one of them.

About Artemis Petconnect, Inc.

Artemis Petconnect, Inc. is the company behind askKynos™, a pet-care platform built on the belief that every pet deserves to be understood. Four equal co-founders, one family, founded the company and built askKynos on nights and weekends: the CEO & President; the CFO & CTO, a PhD data scientist; the CIO & VP Engineering; and the CMO & VP Communications. askKynos offers five kinds of care (pet daycare, boarding, drop-in visits, house sitting and dog walking) built around one persistent story for each pet.

Company: https://www.artemispetconnect.com · Product: https://www.askkynos.ai

Further reading from Artemis Petconnect

- **Eating Our Own Dog Food, Part 3.** The story behind this paper: how we taught askKynos to be understood by AI assistants. https://www.artemispetconnect.com/news/eating-our-own-dog-food-part-3
- **Paper I: The New Economics of Complexity.** How four people built askKynos with AI. https://www.artemispetconnect.com/insights/the-new-economics-of-complexity
- **Paper II: Every Pet Deserves to Be Understood.** Why a pet's story should carry forward. https://www.artemispetconnect.com/insights/every-pet-deserves-to-be-understood

References

[1] Stanford Institute for Human-Centered Artificial Intelligence. (2026). *AI Index Report 2026: Technical Performance.* https://hai.stanford.edu/ai-index/2026-ai-index-report/technical-performance

[2] Li et al. (2026). ComplexMCP: Evaluation of LLM Agents in Dynamic, Interdependent, and Large-Scale Tool Sandbox. *Proceedings of ICML 2026.* https://proceedings.mlr.press/v306/li26ij.html

 © 2026 Artemis Petconnect, Inc. · CC BY-ND 4.0

[3] Liu, Z., et al. (2026). MCPAgentBench: A Real-world Task Benchmark for Evaluating LLM Agent MCP Tool Use. *Proceedings of REALM 2026.* https://aclanthology.org/2026.realm-1.51/

[4] Model Context Protocol. (2026). *Specification, version 2026-07-28.* https://github.com/modelcontextprotocol/modelcontextprotocol/blob/main/docs/specification/2026-07-28/basic/index.mdx

[5] Soria Parra, D., & Delimarsky, D. (2026, July 28). *The 2026-07-28 Specification.* Model Context Protocol Blog. https://blog.modelcontextprotocol.io/posts/2026-07-28/

[6] U.S. National Security Agency. (2026, May 20). *Model Context Protocol (MCP): Security Design Considerations for AI-Driven Automation.* https://www.nsa.gov/Press-Room/Press-Releases-Statements/Press-Release-View/Article/4496698/

Publication details

Author: Artemis Petconnect, Inc. · **Title:** Teaching Applications to Be Understood · **Series:** Artemis Petconnect Insights, Everything Begins with Understanding · **Paper:** III · **Published:** November 2026 · **Address:** https://www.artemispetconnect.com/insights/teaching-applications-to-be-understood

How to cite: Artemis Petconnect, Inc. (2026, November). *Teaching Applications to Be Understood.* Artemis Petconnect Insights, White Paper III. https://www.artemispetconnect.com/insights/teaching-applications-to-be-understood

License: © 2026 Artemis Petconnect, Inc. This paper is licensed under the Creative Commons Attribution-NoDerivatives 4.0 International License (CC BY-ND 4.0): you may share and republish it unchanged, including commercially, with credit to Artemis Petconnect, Inc. https://creativecommons.org/licenses/by-nd/4.0/ The Artemis Petconnect and askKynos names and logos are not covered by this license. ChatGPT, Claude, Grok and Rive are trademarks of their respective owners; no partnership is implied. Figures are conceptual illustrations. Cover illustration created with ChatGPT and edited by Artemis Petconnect.