ACES Consortium - General Overview

Appliance
Communications and
Execution
Standard

ACES Consortium

A cross-domain initiative defining and stewarding an open, vendor-neutral standard for physical and logical interoperability, including power, data, and execution, across modular robotics and IoT systems.

The Challenge

The Core Problem: A Systemic Interoperability Failure

Modern robotic and IoT systems are assembled from an increasing number of heterogeneous components, often sourced from different vendors and industries, each with its own physical interfaces, power requirements, communication protocols, and configuration models.

Interoperability today is achieved through bespoke integration, an approach that does not scale.

- Custom wiring and connectors.
- Vendor-specific adapters and gateways.
- Manual configuration and glue code.
- Tight coupling between hardware and control logic.

As systems grow more modular and autonomous, integration effort grows non-linearly. Each new module increases complexity, cost, and fragility, making systems harder to evolve, maintain, and repurpose over time.

The lack of a shared, vendor-neutral standard for power, data, and execution at the physical interface level has become a systemic bottleneck, not a tooling inconvenience.

Why Fragmentation Is Increasing, Not Decreasing

Rising System Complexity Without Shared Interfaces

Integration complexity scales faster than capability

As capability increases, integration effort grows even faster. Ad-hoc interfaces that work for a single deployment do not generalize across systems, domains, or vendors.

Each new module, sensor, or subsystem introduces new assumptions, new failure modes, and new custom integration work.

System requirements are changing

Modern systems increasingly rely on autonomous decision-making, high-bandwidth sensing, and real-time data processing. AI workloads, advanced sensors, and edge computing are no longer optional features; they are becoming baseline requirements.

At the same time, systems are becoming more modular, distributed, and mobile, with components that must be added, replaced, or upgraded over time.

Each domain reinvents the same solutions

Because there is no shared execution and interoperability layer, each domain solves these problems independently. Similar connectors, protocols, and abstractions are rebuilt repeatedly, often with slight variations that prevent reuse or interoperability.

What is ACES?

ACES - Appliance Communications and Execution Standard

The Open Standard for Modular Robotics and IoT Interoperability

- ACES defines a standardized physical and logical interface that allows modular devices to interoperate without bespoke integration.
- It standardizes how modules connect mechanically, receive power, exchange data, and declare their capabilities.
- ACES enables any compatible controlling system to discover modules and invoke their functions using a common protocol.
- The standard is designed to be industry-agnostic and applicable across agriculture, smart cities, and industrial automation.

What ACES Is Not

ACES does not:

- Define system behavior, orchestration logic, or decision-making.
- Impose how systems are optimized, sequenced, or coordinated.
- Create or require a centralized control layer.
- Dictate how applications manage, orchestrate, or govern systems.
- Replace, compete with, or subsume existing standards or software frameworks.
- Create a closed platform, walled garden, or dependency on any single vendor or consortium member.
- Collect, own, or monetize operational data.

ACES deliberately operates below higher-level software, analytics, and control architectures.

Why These Boundaries Exist

Clear non-goals are as important as goals.
They protect the standard from scope creep and ensure it remains open, inspectable, and implementable by any party. This separation of concerns is intentional.

- It ensures long-term flexibility and technical longevity.
- It allows ACES to coexist with existing and future software ecosystems.
- It prevents lock-in and reduces strategic risk for adopters.
- It enables participation without exclusivity or abandonment of current platforms.
- It preserves neutrality, which is essential for trust and broad, cross-domain adoption.

How ACES Works

High-Level Flow

- Each ACES-compliant module connects to a system via one or more standardized adapters.
- Upon connection, the module exposes an Electronic Data Sheet (EDS) describing its capabilities and requirements.
- A controlling system reads the EDS, determines how the module can be used, and invokes its functions accordingly.
- Data and telemetry generated by the module flow through the adapter to the appropriate consuming system.

Electronic Data Sheet (EDS)

- The Electronic Data Sheet is a machine-readable description published by each module.
- It defines the functions the module exposes, the parameters those functions accept, and the telemetry produced.
- The EDS also specifies power requirements and operational characteristics.
- The EDS communicates what a module can do, leaving all decisions about usage to the controlling system.

How ACES Works

Power and Scaling Model

- ACES is designed for horizontal scaling rather than vertical scaling.
- If a module requires more power, bandwidth, or structural support than a single adapter can provide, additional adapters are added in parallel.
- This approach avoids oversized connectors and preserves modularity.
- Power, data, and mechanical load scale together in a predictable and robust way.

Adapter – Canonical Definition

- The ACES adapter is a passive, declarative interface between a module and a system.
- It provides a standardized mechanical structure, electrical interface, and data transport.
- The adapter contains no intelligence, executes no logic, and makes no decisions.
- Its role is to expose capabilities and carry power and data, not to interpret or control behavior.

How ACES Works

Data Flow Model

- ACES ensures that data generated by a module can flow reliably to the system that invoked the module's function.
- It does not define where data is processed, stored, or analyzed.
- Data routing, aggregation, and optimization are responsibilities of higher-level software frameworks.
- ACES remains agnostic to analytics, orchestration, and application-layer concerns.

Where Does Intelligence Live?

- All intelligence in an ACES-based system exists outside the adapter.
- Control logic may reside in a dedicated edge computing module, a communications module, or a remote system operated by a human or AI.
- ACES places no constraints on where intelligence is located or how it is implemented.
- This allows systems to evolve from simple to highly autonomous without changing the physical interface.

Existing Standards

Existing Standards Are Necessary yet Insufficient

A Structural Gap at the Edge of Autonomous Systems

What existing standards are optimized for

Many established standards successfully address low-speed control, signaling, and coordination within well-defined system boundaries. Others focus on high-throughput backbones and fixed infrastructure designed for heavy, long-lived installations.

These standards have enabled interoperability within their intended contexts and should be understood as foundational to many modern

Where those optimizations break down

Lightweight, modular, and mobile systems place different demands on interfaces. They require compact form factors, combined power and data delivery, dynamic discovery, and predictable execution semantics across components.

These requirements sit between traditional control standards and heavy infrastructure backbones, and are not fully addressed by either.

The resulting gap

The result is a missing interoperability layer at the edge, where autonomous modules, sensors, and subsystems must connect, operate, and evolve across vendors and domains.

ACES is designed specifically to address this gap, without displacing or competing with standards that already serve their domains well.

Position Relative to Existing Standards

Deliberately excludes orchestration and optimization.
Decision-making, routing, and system logic remain the responsibility of higher-level frameworks.

Is orthogonal to messaging and transport protocols.
It does not replace MQTT, HTTP, CAN, ISOBUS, Ethernet, or RF links; it enables modular execution semantics that can be carried over them.

ACES is a boundary standard for modular systems. It standardizes both the physical interface and the execution semantics that enable true plug-and-play modules.

It defines the module boundary.
At that boundary, ACES specifies:
how a module physically connects (power, data, mechanical coupling),
how its capabilities are declared,
how its functions are invoked and monitored.

Ultimately ACES standardizes the physical and logical module boundary to allow modular plug-and-play across disparate systems.

Enables hardware modularity across vendors.
Any ACES-compatible module can be physically connected and logically invoked without bespoke integration, enabling substitution, composition, and scaling.

It sits between orchestration frameworks and transport technologies.
Frameworks express intent and logic; protocols and transports move messages and signals; ACES binds intent to physical capability through a standardized module interface.

ACES as a Standardized Execution Boundary

ACES defines a standard at the module interface, focused strictly on execution.

ACES standardizes:

- a physical plug-and-play module interface,
- capability declaration (what a module can do),
- standardized function invocation,
- execution semantics (acknowledgements, state, errors, telemetry).

ACES explicitly does not define:

- orchestration or decision-making logic,
- optimization, sequencing, or policy management,
- messaging, routing, or transport protocols.

Those concerns remain with existing frameworks and standards.

This separation of concerns allows ACES to coexist with, rather than replace, existing ecosystems.

Existing Technology Domains in Autonomous Systems

Modern autonomous and cyber-physical systems already rely on mature standards and frameworks across multiple domains:

- Control and orchestration frameworks express intent, policies, and logic
  - Examples: FIWARE, LF Edge components, PLC logic, AI systems
- Messaging and transport protocols carry commands and data
  - Examples: MQTT, HTTP, CAN-FD, ISOBUS, Ethernet, RF, cellular
- Physical connectors and interfaces provide power, data, and mechanical coupling
  - Examples: industrial connectors, fieldbuses, vendor-specific interfaces

Each of these domains is necessary and well developed. None of them is sufficient on its own to enable modular, plug-and-play physical systems.

The Missing Standard: Software-to-Physical Execution

Between orchestration software and physical modules lies a critical boundary:

- The point where a software command must be executed by a physical device.

Today, this boundary is typically implemented using:

- proprietary SDKs and APIs,
- custom drivers and firmware,
- bespoke electrical and mechanical interfaces,
- vendor-specific integration logic.

As a result:

- modules are tightly coupled to specific vendors and software stacks,
- swapping or upgrading hardware requires re-integration work,
- “modularity” exists at the concept level, not at the execution level.

This execution boundary is the least standardized part of the system, and the main source of integration friction.

End-to-End Flow with ACES in Place

With ACES, a single command follows a clear, modular path:

1. Intent is created in a control system (human, AI, analytics, automation).
2. Orchestration frameworks decide what should happen and when.
3. Messaging and transport protocols carry the command through the system.
4. ACES defines the execution contract at the module boundary.
5. The physical module executes the function and returns status and data.

ACES occupies the previously unstandardized gap between software intent and physical action, enabling modularity, interoperability, and vendor-agnostic system composition.

Case Studies

Real-World Interoperability

One Standard, Many Applications

- With ACES, modules from different vendors can be connected without rewiring or custom integration.
- Capabilities are discovered automatically through the EDS mechanism.
- Systems can be modified incrementally by adding, removing, or replacing modules.
- Interoperability is achieved at the interface level, not at the behavioral or application level.

- Swap out a sensor, a robotic arm, or a power unit—no rewiring, no reprogramming.
- Integrate devices from different vendors, industries, or generations, all recognized on the same bus.
- Scalable from small test benches to full-scale industrial or environmental deployments.
- Designed for reliability, security, and extensibility as new device types emerge.

Use Case: ACES in an Industrial Environment

Context and Problem

In many industrial environments such as stone crushing, recycling, aggregate processing, or on-site material handling, systems are built as tightly coupled, monolithic installations. Crushers, conveyors, screens, sensors, and safety systems are often sourced from different vendors, wired manually, and configured through proprietary interfaces. Any change in capacity, layout, or operational logic typically requires rewiring, reprogramming, downtime, and vendor-specific intervention. This creates high integration costs, slow adaptation, and strong vendor lock-in.

The goal of this use case is to demonstrate how ACES enables a fundamentally different approach: a modular, self-describing industrial system where physical modules can be added, removed, or replaced without redesigning the system, and where orchestration logic can evolve independently of the hardware.

Use Case: ACES in an Industrial Environment

ACES-Based System Architecture

In this industrial scenario, each physical component is implemented as an ACES-compatible module. Examples include a primary crusher, secondary crusher, conveyor segments, vibration screens, load sensors, motor controllers, dust suppression units, and safety interlocks.

Each module exposes:

- A standardized physical interface for power and data.
- An Electronic Data Sheet (EDS) describing its capabilities, commands, telemetry, power profile, and operational constraints.
- A set of standard ACES communication channels for commands, status, telemetry, and configuration.

Modules do not contain decision-making logic beyond local safety and operational constraints. They do not know the broader system context. Instead, they advertise what they can do, how they can be controlled, and what data they can provide.

Use Case: ACES in an Industrial Environment

Orchestration and Execution Flow

A supervisory control system, which may be an industrial PLC, an edge controller, or a remote software system, discovers all connected modules by reading their EDS files over the ACES protocol. Based on this information, it builds a live model of the available system: which machines exist, what capacities they have, how much power they consume, and what telemetry they expose.

When a crushing operation is initiated, the orchestrator issues standardized ACES commands to the relevant modules. For example:

- Start conveyor A at a given speed.
- Set crusher torque limits.
- Enable dust suppression when vibration thresholds are exceeded.
- Pause downstream modules if an upstream sensor reports overload.

Data flows continuously from modules back to the orchestrator via standardized telemetry channels. If a module is replaced, upgraded, or temporarily removed, the system reconfigures itself automatically based on the updated EDS, without rewriting control logic.

Use Case: ACES in an Industrial Environment

Scalability and Reconfiguration

One of the core advantages demonstrated in this use case is horizontal scalability. If a crusher requires more power or throughput than a single ACES adapter can provide, multiple adapters can be connected in parallel, aggregating power, data bandwidth, and mechanical support. From the orchestration perspective, this remains transparent; the module simply declares its updated power and interface profile.

Similarly, capacity can be expanded by adding additional conveyor segments or processing units. As long as they are ACES-compliant, they become immediately available to the system without bespoke integration work.

Use Case: ACES in an Industrial Environment

Outcome and Value

This use case illustrates ACES as an industrial interoperability layer rather than a control system. ACES does not replace industrial automation software; it replaces brittle, proprietary integration patterns. The result is:

- Faster system assembly and commissioning.
- Reduced downtime during upgrades or maintenance.
- Vendor-neutral procurement.
- Long-term system evolution without architectural rewrites.

ACES-Modular Stone Crushing Plant Use Case

Material Feed

ACES-Capable Feeder Modules

Replaceable Sensors & Actuators

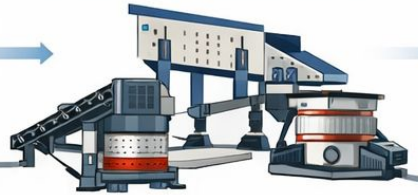
Primary Crushing

ACES-Capable Primary Crusher

Modular Jaw & Motor Units

Screening & Secondary Crushing

ACES-Enabled Screen Units & Secondary Crusher



Interchangeable Modules

Output & Monitoring

Networked Sensors & Telemetry

Final Product and Data

Traditional Plant

- Custom Engineering Required
- Fixed, Hardwired Stages
- High Downtime
- Manual Monitoring & Control

ACES Benefits

- Plug & Play Integration
- Swappable Modules
- Predictive Maintenance
- Improved Efficiency & Flexibility

Use Case: Agrobots Bioromes

Context and Problem

A biorome is a managed ecological system where plants, animals, soil, water, and microclimate interact continuously. Unlike industrial systems, bioromes are dynamic, biological, and non-deterministic. Traditional agricultural machinery is ill-suited to this environment because it is designed for large, uniform operations rather than fine-grained, adaptive intervention.

The challenge is to coordinate many small, specialized robotic and sensing modules across a living landscape, while allowing ecological logic, agronomic expertise, and operational software to evolve independently of the hardware.

Use Case: Agrobots Bioromes

ACES as the Physical and Logical Backbone

In a biorome, ACES acts as the common language and physical interface between heterogeneous modules distributed across the terrain. These modules include, but are not limited to:

- Drone swarms for aerial mapping, seeding, monitoring, and targeted intervention.
- Autonomous ground vehicles with swappable implements for planting, spraying, harvesting, or transport.
- Smart irrigation modules controlling valves, pumps, and emitters.
- Smart pest control modules deploying biopesticides, pheromones, or beneficial insects.
- Fixed environmental sensors measuring soil moisture, nutrients, temperature, humidity, and biodiversity indicators.

Each module is physically connected through one or more ACES adapters and publishes an EDS describing exactly what it can do. Modules have no awareness of “agriculture” as a concept; they simply expose functions, constraints, and telemetry.

Use Case: Agrobots Biomes

LandOS as an ACES-Orchestrating System

LandOS acts as one possible orchestrator of ACES-compatible modules, but it is not a requirement of the standard. In this use case, LandOS discovers all modules in the biorome, ingests their EDS files, and maintains a live representation of available capabilities across the terrain.
Based on ecological models, agronomic rules, human input, and external data (weather, regulations, market constraints), LandOS issues standard ACES instructions to modules. For example:

- Instruct a drone swarm to perform multispectral imaging over a stressed zone.
- Command irrigation modules to reduce flow in areas where soil moisture exceeds thresholds.
- Trigger pest control modules only in zones where sensor data and ecological models indicate risk.
- Coordinate multiple ground vehicles to avoid soil compaction and ecological disturbance.

Crucially, LandOS does not need to know how a specific vendor's irrigation valve works or how a drone stabilizes itself in flight. It only needs to know which commands are exposed via ACES and what telemetry is returned.

Use Case: Agrobots Biomes

Distributed, Adaptive Operation

Because all modules communicate through ACES, the system remains resilient and adaptive. If a module fails, is replaced, or is upgraded, its replacement publishes a new EDS and becomes available immediately. If a new type of module is introduced, such as a novel pest control mechanism or a new sensor, it integrates without changes to existing hardware.

Power and data demands scale naturally. High-demand modules can be provisioned with multiple adapters. Low-power sensors can operate with minimal infrastructure. The same ACES rules apply across the entire biorome.

Use Case: Agrobots Bioromes

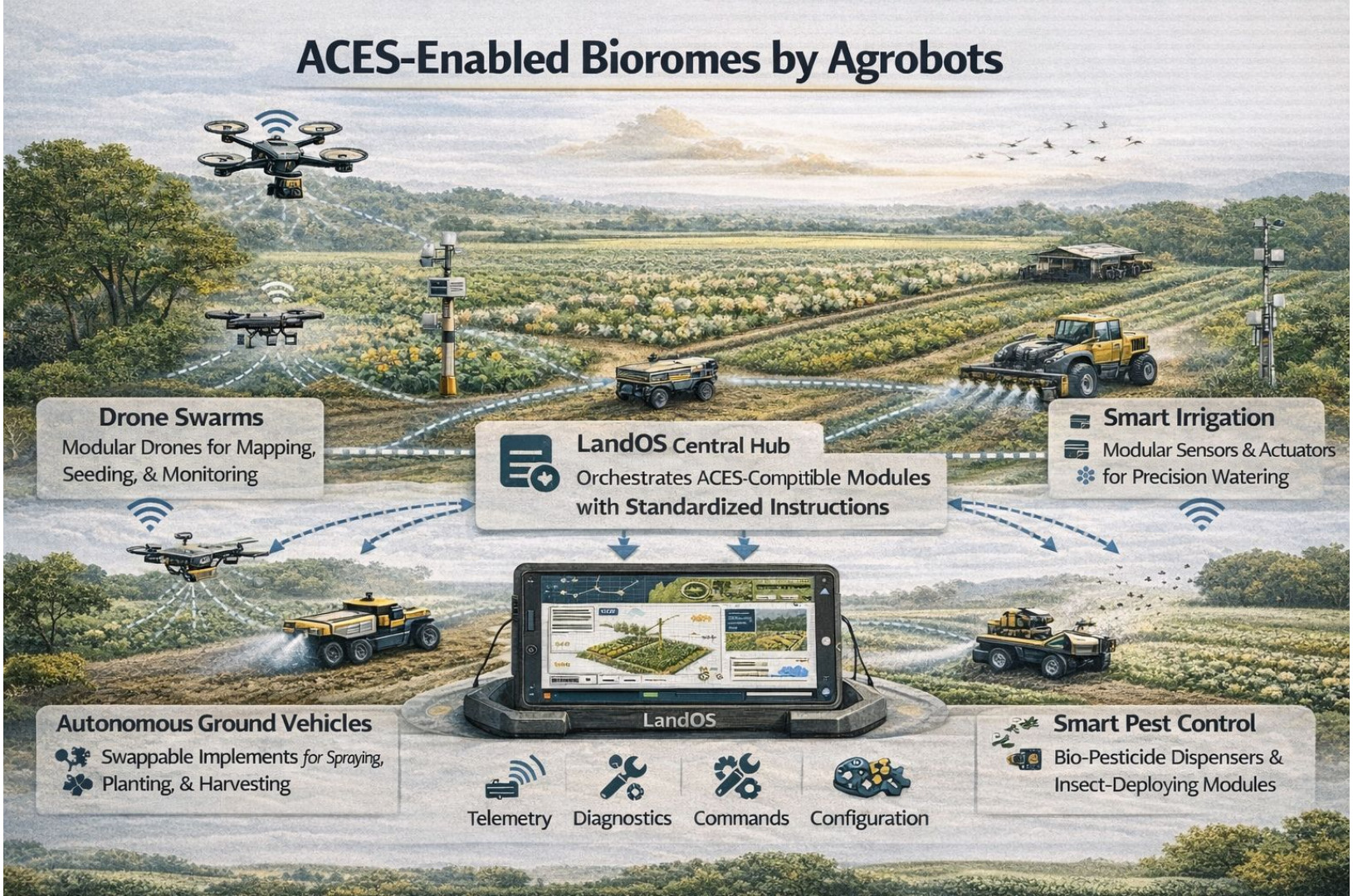
Outcome and Value

This use case demonstrates ACES operating in a biological, non-industrial context, where adaptability and modularity are essential. The result is:

- Fine-grained, targeted interventions instead of blanket treatments.
- Reduced chemical, water, and energy usage.
- A system that evolves over time as ecological understanding improves.
- Complete decoupling between hardware vendors and orchestration logic.

The biorome becomes a programmable ecosystem, not through centralized intelligence embedded in machines, but through standardized execution of modular capabilities exposed via ACES.

ACES-Enabled Bioromes by Agrobots



The Consortium

Strategic Path for ACES Development

ACES Protocol Foundation

- Non-profit association
- ACES protocol open and royalty-free
- Revenue indirect, via members' products
- Broad adoption and ecosystem growth

- **Shared mission:** establish ACES as interoperability standard
- **Same target domains:** agriculture, smart cities, industry
- **Development and Governance:** led by consortium partners

ACES Consortium - Why, How and What

Purpose

To drive standardization in IoT and robotics, creating the conditions for seamless integration, ecosystem growth, and greater value for producers, integrators, and users.

Mission

To co-develop, govern, and maintain the standard. Members collaborate on specifications, certification processes, and ecosystem development and commit to adopting ACES within their own products.

Vision

Create a universal standard for physical interoperability of modular drones, robots, and IoT devices.

Target Applications

- Agriculture 
- Smart Cities 
- Industry 

ACES: Appliance Communications and Execution Standard

Consortium Structure and Sustainability - Open Standard

ACES as a non-profit consortium

The ACES consortium is established as a non-profit entity whose sole mandate is to define, maintain, and steward the ACES interoperability standard.

The consortium itself does not commercialize products, operate platforms, or compete with its members. Its role is to provide neutral governance, technical continuity, and trust in the standard over time.

How members create value

Consortium members benefit from ACES through their own products and services. By building ACES-compatible hardware, software, or systems, members reduce integration cost, expand addressable markets, and participate in a broader interoperable ecosystem.

Commercial value is realized by members independently, through adoption of the standard, not extracted by the consortium.

How the consortium sustains itself

The consortium's revenue is limited and purpose-driven. It may include:

- Research and innovation grants
- Certification fees for non-members seeking ACES compatibility
- Consulting services for non-members adopting the standard

These mechanisms support the operation and evolution of the standard without creating dependency, lock-in, or misaligned incentives.

Why an Open Standard

- Closed connectors and proprietary interfaces benefit individual vendors but fragment the ecosystem.
- An open, vendor-neutral standard lowers integration costs and accelerates innovation.
- ACES enables competition to focus on capabilities, performance, and quality rather than lock-in.
- This shift benefits producers, integrators, and end users alike.

Cross-Domain Consortium

Interoperability Requires Neutral Stewardship

The problem is shared, not domain-specific

Interoperability challenges recur across robotics, sensing, energy, mobility, and infrastructure systems. While implementations differ, the underlying interface problems are the same.

Solving these issues within a single domain or product line inevitably reintroduces fragmentation elsewhere.

The value of a cross-domain consortium

A consortium provides shared governance, pooled investment, and institutional legitimacy. Cross-domain participation strengthens the standard by preventing single-use bias and ensuring that design decisions generalize beyond one application area.

Standards require neutrality and continuity

For an interoperability standard to be trusted, it must be governed independently of any single vendor, product, or commercial roadmap. It must also outlive individual technologies and companies.

This level of neutrality and continuity cannot be achieved by a single organization acting alone.

Interoperability as an Operational Advantage

For system builders and engineers

With a shared execution and interoperability layer, engineers can design modules against predictable interfaces. Integration paths are reduced, assumptions are explicit, and components can be reused across platforms and domains.

This shortens development cycles, lowers integration risk, and allows teams to focus on system capability rather than glue code.

For operators and system users

Interoperable systems allow components to be added, replaced, or upgraded without redesigning the entire system. Failures are easier to isolate, downtime is reduced, and long-term evolution does not require vendor lock-in.

Systems become maintainable assets rather than fragile, one-off deployments.

- Adopt a royalty-free, open-source standard that’s vendor-neutral and ready for any robotics or IoT application.
- Design once, deploy everywhere: ACES modules run out-of-the-box in industrial, commercial, and agricultural environments.
- Slash integration and support costs with a single interface and a quick certification process, provided by members of the consortium.
- Retain full control of your IP while gaining global visibility through the ACES catalogue.

Open European Precedents

KNX Association

(Belgium)

Full Success

- Founded in 1999 as non-profit cooperative (cvba)
- Published royalty-free building automation standard (EN 50090, ISO/IEC 14543)
- 500+ manufacturer members, 100,000+ certified installers in 172 countries

Lesson: open governance sustained global adoption and long-term ecosystem

LF Edge

(Global, EU Members)

Partial Success

- Open foundation for edge computing (part of Linux Foundation)
- Backed by major vendors
- Faced issues with governance complexity, uneven contributions, reliance on few core actors

Lesson: open foundations can stall if funding and community engagement are fragile

FIWARE Foundation

(Germany)

Gaining Traction

- Established 2016 as non-profit e.V.
- Provides open APIs (NGSI-LD) and Smart Data Models for IoT
- Adopted by 350+ cities, growing developer ecosystem

Lesson: open software standard is expanding across smart cities and agriculture

What Makes The ACES Consortium Strong

Digital-enabled business models

Tech-enabled operation serving the traditional sector via a low friction interaction with end-customer

Compelling value proposition

Product offered is 10x better experience than Status quo

Blue-chip management team

Mission-driven entrepreneurs and industry veterans with deep industry knowledge and operational focus

Massive market opportunity

Large Total Addressable Market permits ample room to run and scale

The ACES consortium brings together organizations operating across multiple domains of autonomous and field-deployed systems, each with direct experience building, deploying, and maintaining real-world infrastructure.This combination creates a rare alignment between technical depth, operational reality, and market relevance.

Scalability and geographic expansion

Product, platform and brand with ability to swiftly enter adjacent markets and benefit from economies of scale

High barriers to entry and non-cash moat

Ability to create network effects, economies of scale, or other dynamics that lead to the development of sustainable "moats"

Strong unit economics and cash flow driven

Sound and sustainable unit economics with a prudent balance between profitability and top line growth

Ability to generate a suitable risk-adjusted return

Prudent entry valuation and reasonable path to exit to generate an attractive return for our investors

Governance

ACES Foundation

Legal form

- Non-profit association
- AISBL, e.V.

Governance

- Member-based
- One-member-one-vote
- Committee-driven

Revenue streams

- Consulting
- Certification
- ACES-compatible products and services

Strategic benefits

- Broad adoption
- Credibility (public sector)
- Ecosystem growth
- Faster innovation

IP

- ACES protocol Open Source
- Permissive license
- IP owned by foundation

Main risks

- No direct IP revenue
- Competitors can leverage ACES
- Slower decision-making

From Strengths to Execution

Strength alone is not sufficient

Strong technical teams, market access, and domain expertise create potential, but they do not automatically result in a functioning standard. Without coordination, discipline, and clear ownership, strengths remain fragmented.

ACES exists to convert distributed capability into shared execution.

What execution means for ACES

Execution, in this context, means producing a standard that is clear, implementable, and adopted in real systems. It means reference implementations that work across domains, certification processes that reduce risk, and governance that enables timely decisions.

The measure of success is not agreement, but deployment.

Shared responsibility

Participation in the consortium implies more than alignment. It implies contribution, accountability, and a willingness to converge on shared solutions even when local preferences differ.

The value of ACES depends on collective follow-through.

Why Consortia Fail - Risks We Must Actively Avoid

Ambiguous governance

When decision authority is unclear or contested, standards efforts stall. Technical progress becomes blocked by process debates, and informal power dynamics replace explicit rules.

IP and licensing conflicts

Unclear intellectual property rules create hesitation among contributors and adopters. Fear of future capture or retroactive restrictions undermines trust and participation.

Uneven contribution

When effort and cost are not shared transparently, active members burn out and passive members free-ride. Over time, this erodes motivation and execution capacity.

Slow or blocked decisions

Standards that require unanimity or lack escalation paths become unable to adapt. Delays accumulate, relevance fades, and parallel solutions emerge outside the consortium.

Governance Principles

Designing for Trust, Speed, and Longevity

Neutral stewardship

ACES governance must be institutionally neutral. No single vendor, product line, or domain should be able to steer the standard to its exclusive advantage.

Neutrality is a prerequisite for trust and broad adoption.

Clear IP and licensing posture

All contributions to ACES must be governed by clear, upfront intellectual property and licensing rules. Contributors and adopters must know what they can build, ship, and commercialize without future ambiguity.

Open source licensing is used to reduce friction, not to blur ownership.

Separation of technical and commercial tracks

Technical decisions should be made based on interoperability, implementability, and correctness, not commercial positioning. Commercial collaboration, funding, and go-to-market coordination must be handled in separate, transparent forums.

This separation prevents conflicts of interest from blocking technical progress.

Explicit decision processes

Decision-making authority, escalation paths, and timelines must be defined in advance. Not every decision requires consensus, but every decision must be accountable. Speed and clarity are as important as inclusiveness.

Roadmap

Kick-off to Beta: A Milestone-Driven Roadmap

Joint Development Without Centralized Staffing

How ACES is developed

ACES is developed collaboratively by consortium members, alongside their existing commercial and research activities. Members are not expected to pause their core work or dedicate full-time personnel to ACES.

Instead, development is structured around clearly defined milestones, with bounded scope and shared responsibility.

Roadmap principles

- Progress is measured by deliverables, not hours
- Contributions are distributed across members according to their strengths
- Scope is locked early and expanded deliberately
- External validation is used to unlock leverage

36-month horizon

The objective of this roadmap is to deliver a beta version of the ACES standard that is structurally sound, technically usable, and credible to external stakeholders, including funding bodies and early adopters.

Milestone 1

Institutional-Ready ACES v0 Definition

Timeframe

Y1H1

Deliverables

- Explicit definition of ACES v0 scope and non-scope
- Core cross-domain use cases selected
- Initial technical architecture and boundary definitions
- Public technical and impact documentation suitable for authorities and funding bodies submitted

Milestone 2

Consortium Formation and Governance Lock-In

Timeframe

Y1H1

Deliverables

- ACES non-profit legal entity established
- Consortium charter approved
- Governance structure, roles, and decision processes finalized
- IP contribution and licensing framework locked
- Membership categories, rights, and obligations defined

Milestone 3

Core Interface and Semantics Definition

Timeframe

Y1H2

Deliverables

- Definition of core ACES interfaces (power, data, execution)
- Device discovery and self-description model specified
- Capability declaration and error-handling semantics defined
- Draft ACES core specification

Milestone 4

Reference Implementations and Validation

Timeframe

Y2H1

Deliverables

- Limited set of joint, cross-domain reference implementations
- Validation of ACES interfaces against real systems
- Implementation guidelines and best practices
- Specification refinements based on implementation feedback

Milestone 5

Compliance and Certification Framework (Draft)

Timeframe

Y2H2

Deliverables

- Definition of ACES compliance criteria
- Draft certification levels and requirements
- Test strategy and tooling outline
- Certification pathway defined

Milestone 6

ACES Beta Specification Release

Timeframe

Y3H1

Deliverables

- ACES Beta specification published
- Stable interface guarantees defined
- Backward-compatibility expectations documented
- Beta documentation for implementers

Milestone 7

Beta Validation and Early Deployments

Timeframe

Y3H1

Deliverables

- Beta deployments by consortium members
- Selected beta adopters onboarded
- Certification process exercised in practice
- Consolidated feedback and post-beta roadmap recommendations

Milestone 8

Post-Beta Stabilization and Adoption

Timeframe

Y3H2

Deliverables

- ACES v1.0 stabilization roadmap defined
- Formal certification program launched
- Non-member adoption via certification and consulting
- Expansion of scope driven by validated demand
- Alignment with additional standards bodies and institutions

Resourcing and Financial Sustainability

What requires resourcing

Across the kick-off to beta roadmap, ACES requires focused resourcing in a limited number of areas:

- Consortium coordination and administration
- Legal, governance, and IP documentation
- Core technical specification and reference work
- Compliance, certification, and documentation preparation

These efforts enable member contributions rather than replacing them.

How this is sustained

The consortium's financing model is deliberately constrained and mission-aligned:

- Institutional and research funding, supporting early milestones and standard definition
- Certification fees for non-members, once the standard reaches beta
- Consulting and adoption support for non-members, post-beta

Members realize value through their own products and services, not through consortium revenue.

Why this model works

This model preserves neutrality, avoids lock-in, and aligns incentives. The consortium remains focused on stewardship of the standard, while members and adopters capture commercial value independently.

Financial sustainability supports continuity of the standard, not expansion of control.

External Positioning and Adoption

ACES as neutral infrastructure

ACES is positioned externally as an open, vendor-neutral interoperability standard. It is not a product, a platform, or a commercial offering, but shared infrastructure designed to reduce fragmentation across autonomous and field-deployed systems.

Relationship to existing standards and ecosystems

ACES is explicitly complementary to existing standards. It does not seek to replace domain-specific protocols or infrastructure layers, but to provide a missing execution and interoperability layer at the edge.

This positioning avoids duplication, conflict, and unnecessary standard proliferation.

Adoption and use

Adoption of ACES is voluntary and driven by implementability. External organizations may adopt ACES through direct implementation, certification, or supported adoption, without becoming consortium members.

There are no mandates, exclusivity requirements, or gatekeeping mechanisms.

Thank You!