

System Architecture Overview

This page defines the **AOFS system architecture**, including all layers, their responsibilities, interactions, and data flows. It establishes **authority boundaries**, **offline operation rules**, and **federation/synchronization requirements** for AOFS controllers.

All AOFS-compliant systems **must conform** to the rules specified here.

1. Architecture Layers

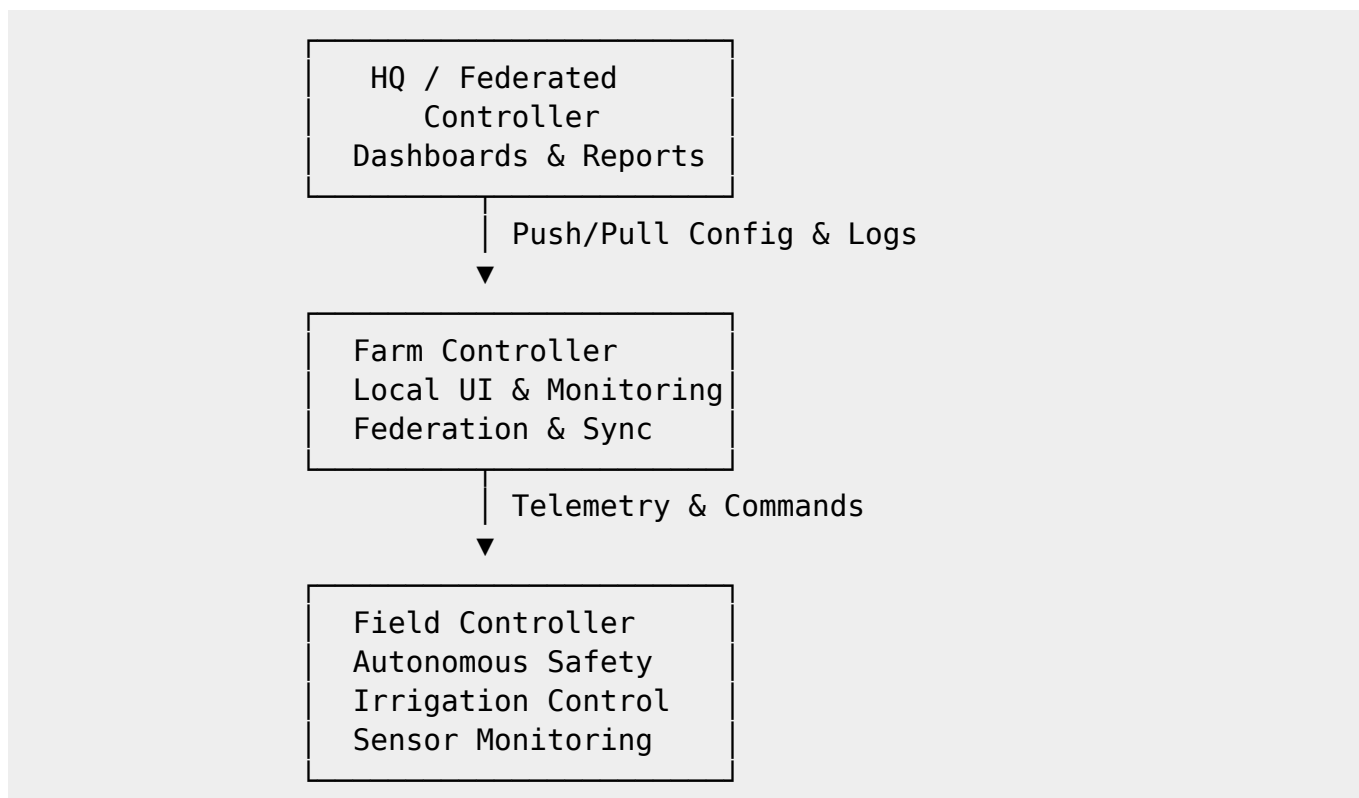
AOFS defines three core layers:

- [Field Controller Layer](#)
- [Farm Controller Layer \(Local / Federated\)](#)
- [HQ / Federated Controller Layer](#)

All controller layers **expose a human interface appropriate to their authority and audience**, while **never bypassing Field Controller safety rules**.

—

2. Authority & Data Flow Diagram



Legend:

- **Field Controller:** authoritative for safety-critical irrigation.
- **Farm Controller:** local supervision, configuration, and federation; respects Field Controller authority.
- **HQ Controller:** multi-farm oversight, reporting, analytics; may propose updates but cannot override Field Controller safety logic.

3. Controller Responsibilities

Layer	Primary Role	Human Interface	Connectivity	Authority
----	-----	-----	-----	-----
Field Controller	Execute irrigation & safety	Embedded UI for monitoring / non-critical overrides	None (offline)	Authoritative locally
Farm Controller	Local supervision & federation	Full UI: monitoring, configuration	Optional (for federation)	Supervisory (non-critical only)
HQ Controller	Multi-farm oversight & analytics	Dashboards, reporting, config proposals	Required for federation	Supervisory (proposals only)

4. Federation / Sync Model

- **Push/Pull:** Farm Controllers sync with HQ and/or peer farms.
- **Conflict Resolution:**
 - Timestamp precedence
 - Operator approval for schedule/config conflicts
 - Field Controller safety rules **always take priority**
- **Offline First:**
 - Controllers continue autonomous operation if disconnected
 - Logs and changes queue for synchronization once connectivity is restored

5. Human Interface Rules

- All controllers expose interfaces appropriate to their role:
 - Field Controller: embedded status UI, safety alerts, non-critical operator overrides
 - Farm Controller: full local UI for monitoring, configuration, and federation
 - HQ Controller: multi-farm dashboards, analytics, authorized configuration proposals
- No interface may bypass Field Controller safety rules.

6. Communication Model & Protocol Independence

- Controller layers may communicate using one or more standardized protocols (see [Communication Protocols & Standards](#))
- Protocol choice does not define authority
- Communication transport is strictly separated from control authority
- Field Controller authority is defined by architectural rules, not by message origin
- Remote commands received via MQTT, AMQP, or other protocols must always be validated locally
- Loss of connectivity must never affect safety-critical irrigation execution
- Communication failure must default to safe autonomous operation
- AOFS architecture is protocol-agnostic
- Multiple protocols may coexist within a deployment
- Implementations must ensure auditability of all received and transmitted messages

7. Compliance Notes

- AOFS-compliant deployments **must implement all three layers** as defined.
- Field Controller safety rules **cannot be overridden** by higher layers.
- All push/pull, configuration changes, and operator actions **must be logged**.
- Offline operation **must not compromise irrigation or safety**.
- Failure to respect authority boundaries **invalidates AOFS compliance**.

8. References

- [Field Controller Layer](#)
- [Farm Controller Layer \(Local / Federated\)](#)
- [HQ / Federated Controller Layer](#)
- [Hydraulic & Water Systems](#)
- [Electrical & Control Interfaces](#)
- [Measuring, Monitoring & Documentation Systems](#)

From:

<http://wiki.irrigation.afriticgroup.com/> - **Afritic Open Farming Standard**

Permanent link:

<http://wiki.irrigation.afriticgroup.com/doku.php?id=architecture:start>

Last update: **2026/02/23 01:11**

