

ORIGINAL RESEARCH

Open access

Design Principles for Low-Resource Smart Farming Platforms: A Design Science Framework for Smallholder and Connectivity-Constrained Agriculture

Lina Hassan^{1*}, Omar Khalaf², Reem Jaber¹, Rania Mostafa³

Abstract

Smart farming platforms are frequently conceived around persistent connectivity, modern devices, scalable cloud services, specialized technical support, and the financial capacity to adopt integrated digital systems. These assumptions may not hold in smallholder and connectivity-constrained agricultural settings, where infrastructure gaps, uneven digital skills, limited advisory access, maintenance difficulties, data costs, language diversity, and uncertain platform governance shape whether digital services are usable or sustainable. This article develops a conceptual design science framework for low-resource smart farming platforms. The framework links problem relevance, contextual requirement definition, design principles, platform artifact logic, implementation conditions, evaluation needs, feedback, and framework updating. It focuses on affordability, transparent costs, incremental adoption, modular architecture, interoperability, offline-first operation, low-bandwidth communication, local usability, accessible interaction, advisory integration, maintenance capacity, farmer data control, and trust. The proposed approach treats smallholders as heterogeneous users operating within diverse production systems, infrastructures, institutions, and knowledge networks. Platform functions are therefore organized as configurable responses to identified conditions rather than as a universal technology package. The principal contribution is a structured design science logic for translating low-resource agricultural problems into bounded platform requirements and evaluable conceptual artifacts. The framework can guide platform design and research planning, but it does not independently demonstrate affordability, inclusion, adoption, scalability, sustainability, operational safety, or deployment readiness. These outcomes require local usability assessment, affordability analysis, governance review, maintenance evaluation, implementation research, and field validation involving intended users and supporting institutions.

Keywords Smallholder agriculture, Affordability, Smart farming platforms, Design science, Low-resource agriculture, Connectivity constraints

*Correspondence:

Lina Hassan
lina.hassan@ju.edu.jo

¹ Department of Low-Resource Agricultural Design, Faculty of Agriculture, University of Jordan, Amman, Jordan

² Department of Smallholder Connectivity and Smart Farming, Faculty of Agriculture, Jordan University of Science and Technology, Irbid, Jordan

³ Department of Design Science for Agriculture, Faculty of Agriculture, Al-Balqa Applied University, Salt, Jordan

Introduction

Digital agriculture is increasingly associated with connected sensors, data platforms, decision-support services, mobile applications, remote sensing, automated analytics, and digitally mediated farm management. However, the significance of this transformation cannot be understood only through the availability or technical performance of individual technologies. Digital agriculture also changes work practices, knowledge relationships, data ownership, advisory roles, institutional

responsibilities, and the distribution of decision-making power. These sociotechnical dimensions are particularly important in low-resource agriculture, where technology choices interact with constrained capital, uneven infrastructure, limited technical support, and diverse farmer capabilities. A platform may be technically functional yet remain poorly aligned with local priorities, unaffordable over its operating life, difficult to repair, inaccessible to some intended users, or dependent on institutions that cannot sustain it. Consequently, low-resource smart farming should be treated as a contextual design problem rather than as a simplified version of high-resource precision agriculture [1].

Smart farming platforms also operate within agricultural knowledge and advice networks rather than functioning as isolated information products. Farmers may interpret digital outputs through extension officers, agronomists, cooperatives, input suppliers, local service providers, peer networks, or other trusted intermediaries. Digitalisation can alter how information is produced, validated, transmitted, and acted upon, while also shifting responsibilities among public institutions, private firms, advisers, and farmers. In connectivity-constrained settings, a platform that generates data without providing understandable interpretation or feasible access to human assistance may have limited practical utility. Advisory integration must therefore be considered an architectural and organizational requirement, not an optional feature added after technical development. Platform design should clarify how information moves between digital components and human actors, how recommendations can be questioned or escalated, and how farmer feedback can influence later revisions [2].

A further challenge is that low-resource agricultural platforms combine social, cyber, and physical elements. Their operation depends not only on software and data processing but also on devices, communication networks, energy access, farm environments, institutional arrangements, user practices, and local support capacity. These elements interact across several levels. A mobile interface may be usable but unavailable during network interruptions; a sensor module may collect data but become ineffective when replacement parts are inaccessible; and a cloud service may process information while leaving farmers uncertain about how their data are used. A context-sensitive design process must therefore map the social actors, technical dependencies, physical conditions, and responsibility relationships surrounding the intended artifact. This mapping helps distinguish requirements that can be addressed through platform architecture from constraints that require institutional support, training, financing, governance, or infrastructure development [3].

Design science is appropriate for this problem because it connects consequential problems with explicit requirements, prescriptive design principles, artifact configurations, implementation conditions, and evaluation. It can help structure how low-resource conditions are translated into modular functions, offline workflows, accessible interfaces, governance controls, and maintenance arrangements. Nevertheless, a design science framework does not by itself establish that a platform will be affordable, adopted, inclusive, trusted, scalable, sustainable, or suitable for deployment. Such outcomes depend on farmers' priorities, institutional capacity, infrastructure, economic conditions, social relations, and evidence generated through appropriate evaluation. This article therefore develops a conceptual framework for designing and assessing low-resource smart farming platforms while maintaining explicit boundaries between proposed design logic and demonstrated implementation outcomes. Its contribution lies in organizing context definition, design requirements, platform principles, artifact logic, evaluation, feedback, and updating around smallholder and connectivity-constrained agriculture [4].

Low-Resource Smart Farming Contexts

Low-resource smart farming contexts are characterized by combinations of constrained finance, uneven infrastructure, limited access to specialized equipment, variable digital skills, fragmented advisory services, and restricted capacity to maintain complex systems. These conditions should not be interpreted as a single fixed category. Smallholder farms differ in production orientation, labour organization, market participation, asset ownership, risk exposure, language, education, and access to collective institutions. Digital agriculture in these settings is therefore better understood as an ecosystem involving farmers, communication services, data systems, advisers, public institutions, technology providers,

cooperatives, local technicians, and market actors. Weakness in any of these components can affect the utility of the overall platform. An application may be accessible through a phone but remain unusable because information is poorly localized, costs are unclear, training is unavailable, or the service cannot connect farmers with trusted support [5].

Resource constraints influence both the scale and sequence of digital adoption. Small farms may be unable to justify systems designed around extensive sensing infrastructure, continuous subscriptions, specialized calibration, or high volumes of data. The relevant design question is not whether precision agriculture can be miniaturized, but which functions are valuable enough to justify their full economic and operational burden in a specific setting. This requires attention to initial acquisition, connectivity, replacement, repair, training, technical support, and the time required to enter or interpret data. Incremental adoption may allow users to begin with farm records, weather information, lightweight advisory communication, or limited sensing before considering more complex modules. However, staged access does not guarantee affordability or progression. Each configuration requires separate assessment of utility, recurring cost, maintenance demand, and compatibility with existing farming practices [6].

Physical and communication infrastructure further shapes feasible platform architecture. Smallholder-oriented sensing and Internet of Things systems may depend on network coverage, power availability, durable hardware, local gateways, device storage, or access to cloud services. Where these dependencies are unstable, continuous data transmission and centralized processing may be inappropriate for essential functions. Edge processing, local data storage, delayed synchronization, or intermittent upload can reduce reliance on continuous connectivity where supported by the use case and validated under actual operating conditions. Similarly, sensor integration should remain modular so that platform utility does not collapse when a device fails or cannot be replaced. Maintenance responsibility, diagnostic procedures, component availability, and repair capacity should be specified during design rather than treated as concerns that arise only after installation [7].

Low-resource conditions are also produced by institutional and social arrangements. Government support, advisory capacity, platform business models, technology access, farmer knowledge, collective organization, and local service ecosystems influence whether digital services can be used meaningfully. Platform design must therefore begin with stakeholder and context mapping rather than a predetermined technical package. This includes identifying intended users, intermediaries, infrastructure providers, data stewards, maintenance actors, and groups that may be excluded by device, language, literacy, gender, age, disability, income, or geographic location. It also requires an explicit no-one-size-fits-all boundary. A configuration developed for one smallholder setting cannot be assumed to transfer to another without examining differences in farm systems, institutions, infrastructure, advisory relationships, and data-governance expectations [8].

Smallholder Agriculture and Connectivity Constraints

Mobile access is often treated as the primary route for reaching smallholder farmers, but the existence of mobile networks or household phone ownership does not demonstrate effective access to agricultural services. Devices may be shared, technically outdated, unavailable to particular household members, or unable to support data-intensive applications. Farmers may also face high data costs, limited storage, intermittent charging, unfamiliar interfaces, or concerns about privacy when accounts are accessed through shared devices. Mobile agricultural services should therefore be assessed according to practical utility, accessibility, task completion, information quality, and continuity of use rather than the number of potential users who possess a phone. Lightweight messaging, assisted access, or voice interaction may be appropriate in some contexts, but each channel requires local evaluation of language, comprehension, cost, privacy, and reliability [9].

Digital adoption among small-scale farmers is shaped by interacting economic, technological, individual, organizational, and social conditions. Affordability may affect initial access, while perceived usefulness, digital confidence, training, compatibility with farm practices, and continuing support influence whether a service is used after introduction. Social networks and institutional relationships can also affect how farmers judge the legitimacy or risk of a platform. These

factors caution against assuming that a simple interface or low purchase price will produce adoption. Design can reduce avoidable burdens by limiting unnecessary steps, clarifying expected benefits and costs, and supporting gradual engagement, but structural constraints may remain. Sustained use should consequently be treated as an empirical evaluation question requiring attention to discontinuation, selective use, adaptation, and reliance on intermediaries [10].

Connectivity constraints also affect the relationship between digital platforms and agricultural extension. Digital communication may increase the potential reach of advisory services, but it does not automatically provide locally relevant interpretation, accountability, or responsive support. Farmers may require assistance to translate alerts, forecasts, sensor readings, or model outputs into decisions that reflect crop conditions, resource limitations, and risk. Platforms can support hybrid advisory arrangements by enabling structured referrals, adviser access, shared records, feedback, and escalation when automated or standardized information is insufficient. Such integration depends on the capacity and mandate of extension services, cooperatives, agronomists, and other intermediaries. Where human support is limited, adding a digital channel may redistribute workload rather than resolve the advisory constraint [11].

Intermittent networks, low bandwidth, electricity limitations, restricted device access, data costs, digital literacy, and language diversity collectively delimit which platform functions can be relied upon. Essential tasks may need to remain available without a continuous connection, while non-essential updates can be synchronized later. Interfaces may require text-light, localized, or multimodal alternatives where these are supported by user research. Yet connectivity-resilient design cannot resolve all exclusion risks. Farmers may remain excluded through income, social position, disability, limited control over devices, distrust, or lack of institutional support. Platform dependency and vendor lock-in create further risks when data cannot be exported, functions cannot be transferred, or service continuity depends on a single provider. Low-bandwidth and offline-first operation should therefore be combined with interoperability, understandable data controls, local support, and explicit inclusion and adoption boundaries [12].

Design Science Logic for Agricultural Platforms

Design science distinguishes between the problem being addressed, the requirements derived from that problem, the principles proposed to guide design, the artifact that instantiates those principles, and the evaluation needed to examine utility. A design principle should not be expressed as a broad aspiration such as “make the platform affordable” or “design for farmers.” It should specify the intended action, the mechanism through which the action is expected to address a problem, the users or implementers involved, the contextual conditions under which it may apply, and the boundaries of the resulting claim. For example, a modular architecture may permit components to be introduced or replaced incrementally, but this mechanism does not establish that the overall platform is affordable or scalable. Those properties require separate economic and implementation evidence [13].

Within agricultural systems, design science offers a way to connect complex practical problems with iterative artifact development. The process begins with problem relevance: the proposed artifact should address a consequential need identified in the intended context rather than a technical possibility selected in advance. Contextual evidence is then translated into functional requirements, such as access to farm records or advisory communication, and non-functional requirements, such as offline operation, accessibility, maintainability, privacy, and interoperability. These requirements guide architecture and interface decisions, but they must remain traceable to the problem evidence from which they were derived. Design utility is examined by determining whether intended users can perform relevant tasks under realistic conditions, not merely whether the software executes as specified [14].

Farmer-centered, human-centered, participatory, and co-design approaches can strengthen this logic by involving intended users in problem formulation, requirement definition, prototype interpretation, and revision. Participation is especially important where designers may otherwise rely on assumptions about farmer priorities, literacy, technology access, risk tolerance, or trusted sources of advice. Nevertheless, participation should not be treated as a ceremonial consultation or as proof that the resulting artifact has been approved by farmers. Meaningful co-design requires attention to representation, decision rights, facilitation, time, resources, and the extent to which participant input changes the

artifact. It should also recognize differences among farmers rather than treating the most accessible participants as representatives of all intended users [15].

Artifact evaluation must be staged according to the maturity and claims of the design. Conceptual review can assess the coherence and completeness of requirements and principles. Architecture evaluation can examine modularity, interfaces, synchronization, security, and failure recovery. Usability assessment can examine task completion, comprehension, accessibility, error, and support needs. Implementation evaluation can investigate maintenance, advisory capacity, governance, cost, institutional fit, and sustained use. Field validation is required before claims are made about performance or utility under operational farming conditions. Evaluation findings should feed back into context mapping, requirements, architecture, interfaces, and boundary statements. Accordingly, the framework developed in this article is a conceptual design artifact and a structure for future evaluation; it is not an implemented system, validated platform, universal standard, or deployment-ready architecture [14].

Affordability, Modularity, and Local Usability

Affordability should be treated as a contextual requirement rather than an inherent property of a low-cost device or simplified application. The relevant cost includes acquisition, connectivity, subscriptions, training, maintenance, repair, replacement, data entry, and the time required to obtain and interpret information. These burdens may be distributed differently among farmers, cooperatives, advisers, service providers, and public institutions. Farmer-centered design can help reveal which costs and tasks users regard as acceptable, which functions address meaningful priorities, and which forms of support are necessary. It also requires designers to recognize that farmers can participate as co-designers only when processes, resources, representation, and decision rights enable meaningful influence. Farmer involvement should therefore inform affordability requirements and configuration choices without being interpreted as evidence that a platform is affordable, accepted, or approved by all intended users [16].

Local usability begins before interface production. Designers must understand how farmers describe problems, organize tasks, interpret information, share devices, seek advice, and respond to uncertainty. Empathising, defining, and ideating with farming communities can expose differences between a technically convenient feature and a practically useful workflow. Interface decisions should reflect locally validated language, terminology, reading practices, numeracy, device capability, and accessibility needs. Text-light layouts, icons, audio prompts, voice interaction, or assisted workflows may reduce some barriers where locally appropriate, but each option can introduce new risks, including ambiguous symbols, transcription errors, privacy concerns, and dependence on external assistance. Usability must consequently be examined through representative task completion, comprehension, error recovery, and support requirements rather than visual simplicity alone [17].

Modularity can support incremental configuration by separating farm records, sensing, communication, analytics, advisory services, and external data connections into replaceable or optional components. A smallholder-oriented platform architecture may, for example, permit a farmer or supporting organization to begin with limited data capture and advisory functions before considering sensor or remote-sensing integration. Local processing and edge components may also reduce some dependence on continuous communication. Nevertheless, modularity is valuable only when module boundaries, data formats, interfaces, dependencies, and maintenance responsibilities are clearly specified. A platform described as customizable or cost-effective in one setting cannot be assumed to remain affordable, maintainable, interoperable, or scalable elsewhere. These properties require independent assessment under the intended institutional, economic, and infrastructural conditions [18].

Mobile and lightweight agricultural services further demonstrate that accessibility depends on more than technical availability. Perceived usefulness, ease of use, confidence, training, service reliability, and continuing support influence whether users can integrate an application into farming practices. Offline-first operation may preserve selected tasks during network interruption, while low-bandwidth modes, local storage, SMS, or voice channels may support alternative access where validated. However, these mechanisms can create synchronization conflicts, stale information, incomplete

advice, data-security risks, or additional costs. Maintenance and repairability must also be planned through diagnostic procedures, replaceable components, documented interfaces, local support roles, and realistic access to parts. **Table 1** summarises low-resource context conditions, connectivity constraints, design science logic, affordability requirements, modularity needs, and local usability requirements for smart farming platforms [19].

Table 1. Low-Resource Context Conditions and Design Requirements for Smart Farming Platforms: Smallholder Agriculture, Connectivity Constraints, Rural Infrastructure, Mobile Access, Offline-First Design, Low-Bandwidth Operation, Affordability, Modularity, Interoperability, Local Language Support, Low-Literacy Interfaces, Maintenance, Repairability, and Inclusion Boundaries

Context condition or design requirement	Core design problem	Required contextual understanding	Platform design response	Potential platform output	Failure or exclusion risk	Evaluation need
Smallholder farm heterogeneity	A standardized configuration may not match diverse production systems and priorities	Farm type, crops, livestock, labour, market relationships, risk, and decision practices	Create configurable workflows and context-specific requirement profiles	Locally bounded platform configuration	Misaligned functions and inappropriate recommendations	Relevance assessment across distinct farmer groups
Resource-constrained farming	Platform demands may exceed available finance, time, equipment, or support	Capital availability, seasonal income, labour burden, and existing assets	Define minimum viable functions and optional modules	Staged functional configuration	Financial burden, underuse, or abandonment	Total-cost and operational-burden assessment
Rural infrastructure	Essential services may rely on unreliable electricity, networks, transport, or repair systems	Power access, coverage, charging, hosting, transport, and service availability	Map and minimize critical external dependencies	Infrastructure-aware architecture	Persistent downtime or inaccessible services	Dependency and uptime assessment
Mobile access	Phone ownership may not equal private, continuous, or technically suitable access	Device type, sharing, storage, operating system, privacy, and charging	Support appropriate device diversity and secure shared-device workflows where validated	Mobile or assisted-access channel	Gendered, household, technical, or privacy exclusion	Access and task testing by device and user group

Intermittent connectivity	Continuous online workflows may fail during essential tasks	Outage frequency, duration, latency, and synchronization opportunities	Provide local task completion, queuing, and delayed synchronization	Recoverable offline workflow	Lost transactions, conflicts, or stale records	Disconnection, recovery, and reconciliation testing
Low-bandwidth operation	Data-heavy interfaces may be slow or unaffordable	Throughput, tariffs, device capacity, and task urgency	Use compressed payloads, text-first modes, and deferred media	Low-data communication mode	Incomplete information or unusable response times	Payload, latency, comprehension, and cost assessment
Electricity limitation where supported	Devices and local nodes may not remain continuously powered	Charging access, outage patterns, energy demand, and battery replacement	Use energy-aware operation and low-power components where appropriate	Extended functional availability	Device shutdown, battery degradation, or replacement burden	Energy-use, uptime, and replacement assessment
Affordability	Initial price may conceal continuing and indirect costs	Acquisition, subscription, data, training, support, repair, replacement, and user time	Minimize avoidable costs and permit staged configurations	Context-specific cost profile	Exclusion, debt, discontinuation, or stranded equipment	Full-cost and continuation assessment
Cost transparency	Users may not understand the complete financial commitment	Pricing terms, service changes, connectivity costs, and maintenance responsibilities	Present all foreseeable cost categories in understandable form	Transparent cost statement	Hidden charges and underestimated operational burden	Cost verification and comprehension testing
Incremental adoption	A complete system may require excessive initial commitment	Priority tasks, existing capacity, module independence, and upgrade conditions	Allow limited entry configurations and optional expansion	Staged adoption pathway	Early modules may provide insufficient value or create upgrade pressure	Utility and cost assessment at every stage
Modularity	Monolithic systems are difficult to adapt, repair, or scale down	Functional dependencies, local priorities, and component availability	Separate functions through defined interfaces and replaceable modules	Core and optional platform modules	Hidden coupling or incompatible replacements	Isolation, integration, and replacement testing

Scalability boundary	Modular architecture may not remain viable across sites or user volumes	Infrastructure, support, governance, service capacity, and contextual variation	Document scale assumptions, bottlenecks, and configuration limits	Scalability-condition statement	Service degradation or inappropriate transfer	Multi-site and organizational evaluation
----------------------	---	---	---	---------------------------------	---	--

Platform Design Principles

The first proposed principle is context-first platform design. Platform development should begin with a bounded account of the agricultural problem, intended users, farm diversity, infrastructure, institutions, knowledge networks, and supporting actors. The architecture should then reflect the data flows and decision functions required by this context. Smart farming data systems commonly connect sensing, storage, analytics, service providers, farm management, and decision support, making interoperability and clear actor responsibilities important. In low-resource settings, however, this general architecture must be reduced or reconfigured according to actual utility and capacity. Interoperability should therefore support exchange among justified modules without requiring every available technology, service, or dataset to be integrated [20].

The second principle is accessible modularity with incremental value. Each module should address a defined problem and remain understandable within the farmer's or supporting organization's workflow. Farm records, advisory communication, weather information, sensing, analytics, and market-related functions can be separated conceptually so that their costs, dependencies, and value can be evaluated independently. Data-management research involving smallholders demonstrates the importance of organizing fragmented records and information services around practical user needs. A modular platform may support this organization through shared identifiers, simple records, and bounded decision-support functions. Yet modules should not be added merely because they are technically connectable. Their inclusion must depend on contextual relevance, data availability, maintenance capacity, governance safeguards, and evidence that users can interpret the resulting output [21].

The third principle is farmer-controlled and governance-aware data use. A low-resource platform should make clear what data are collected, why they are required, who can access them, how long they are retained, whether they may be reused, and how farmers can obtain or transfer their records. Farmers' reluctance to share data is not simply a communication problem; it can reflect uncertainty about ownership, control, contractual terms, commercial benefit, and legal protection. Design responses may include data minimization, understandable permissions, access logs, export functions, correction procedures, and differentiated controls for farmers, advisers, cooperatives, and service providers. These mechanisms may improve transparency, but they do not resolve all legal, economic, or bargaining asymmetries. Governance review must therefore accompany interface and architecture design [22].

The fourth principle is maintenance-aware, support-integrated, and feedback-enabled design. Platform transparency should extend beyond data terms to the provenance of advice, limits of automated outputs, responsibilities for maintenance, and procedures for reporting errors. Trust cannot be inferred from system availability or technical sophistication. It is influenced by whether users understand how information and data are handled, whether expected benefits are credible, and whether problems receive accountable responses. The combined principle set therefore links context-first and farmer-centered design with offline-first operation, low-bandwidth communication, affordability transparency, modularity, interoperability, local usability, advisory support, data governance, maintenance, and iterative evaluation. **Figure 1** synthesises the design principles for low-resource smart farming platforms, linking context-first design, offline-first operation, affordability, modularity, local usability, advisory integration, data governance, maintenance, and feedback-enabled improvement [23].

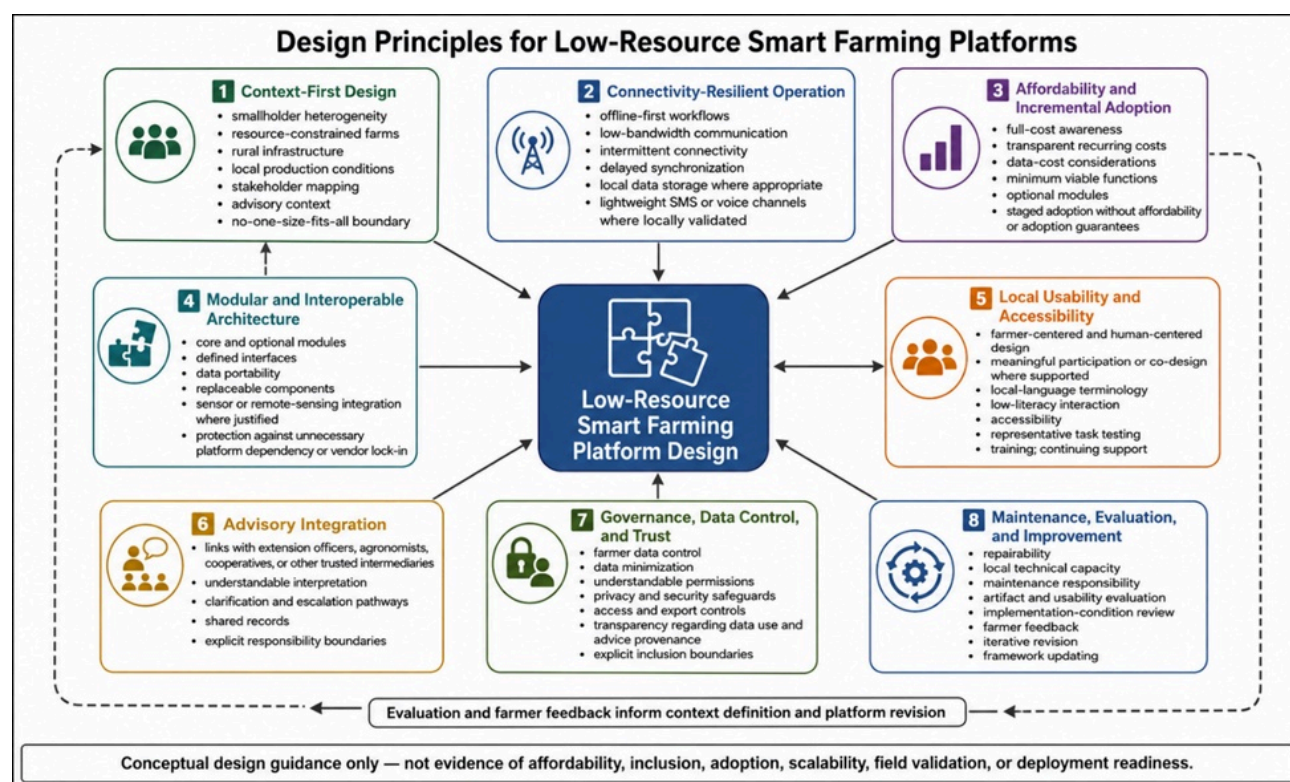


Figure 1. Design Principles for Low-Resource Smart Farming Platforms

Alt-text description:

A conceptual design-principle map showing low-resource smart farming platform requirements such as smallholder context, connectivity constraints, affordability, modularity, offline operation, low-bandwidth communication, usability, advisory integration, data governance, maintenance, and farmer feedback.

Proposed Design Science Framework

The proposed framework begins with low-resource problem definition and contextual mapping. Designers first identify the practical problem, its consequences, and the users or institutions affected without assuming that a digital platform is the necessary solution. Smallholder context mapping then records relevant farm types, livelihoods, production systems, decision practices, languages, skills, device access, infrastructure, advisory relationships, and exclusion risks. Stakeholder mapping identifies farmers, advisers, extension services, cooperatives, platform operators, data stewards, local technicians, and public or private institutions. This contextual layer treats the platform as part of a digital innovation ecosystem in which technical modules, organizational arrangements, business relationships, and governance mechanisms influence one another. Its purpose is to ensure that artifact design remains connected to problem relevance and multi-actor implementation conditions [24].

The second component is constraint assessment and requirement definition. Connectivity and infrastructure assessment records network interruptions, bandwidth limitations, data costs, electricity dependencies, device conditions, hosting arrangements, and maintenance access. Affordability assessment identifies acquisition, recurring, training, repair, replacement, and user-time costs. Digital inclusion assessment considers language, literacy, disability, gender, age, device control, income, and reliance on intermediaries. These findings are converted into traceable design requirements. A reference-architecture approach can distinguish common components from context-dependent variation points, allowing core data, identity, synchronization, and governance functions to be separated from optional sensing, remote-

sensing, advisory, marketplace, or analytics modules. This architectural logic supports adaptation, but it does not establish affordability, interoperability, or scalability until those properties are evaluated in the intended setting [25].

The third component is platform artifact design. The conceptual artifact combines modular architecture, offline-first operation, low-bandwidth communication, local usability, accessibility, advisory integration, data governance, and maintenance planning. Core tasks should be prioritized according to their relevance under disconnected conditions. Local storage and delayed synchronization may support essential records, while lightweight messaging or voice interaction may supplement application-based interfaces where locally appropriate. Advisory integration specifies when information can be used directly, when human interpretation is required, and how users can request clarification or report problems. The design must also identify which modules are unnecessary in a given setting. Evidence concerning digital tools for African smallholders remains heterogeneous, and attribution of farm-level outcomes is often incomplete. The framework therefore requires local validation rather than assuming that a function successful in one context will deliver similar value elsewhere [26].

The fourth component is governance, support, and implementation-condition review. Data collection should be limited to specified functions, and farmer-facing controls should make access, sharing, reuse, correction, retention, and export understandable. Privacy and security requirements should be proportionate to the data and risks involved, while platform dependency and vendor lock-in should be examined through portability, interface documentation, service continuity, and exit arrangements. Critical analysis of agricultural platformization shows that access to a digital platform can coexist with surveillance, data extractivism, unequal benefit distribution, and limited farmer control. Consequently, the framework does not equate digital participation with inclusion. Governance safeguards, local capacity, maintenance arrangements, training, adviser availability, and organizational accountability are required implementation conditions rather than automatic consequences of platform architecture [27].

The final component is artifact evaluation, feedback, iteration, and claim control. Evaluation is staged from conceptual coherence and architecture testing to usability, accessibility, synchronization, security, maintenance, affordability, governance, implementation, and field assessment. Farmer and intermediary feedback should be recorded as traceable evidence for revising context maps, requirements, modules, interfaces, and boundary statements. The design cycle remains open because infrastructure, service providers, costs, user practices, and institutional responsibilities can change. The framework explicitly retains no-affordability, no-inclusion, no-adoption, no-scalability, and no-deployment-readiness boundaries until appropriate evidence is generated. **Table 2** presents the proposed design science framework for low-resource smart farming platforms.

Table 2. Proposed Design Science Framework for Low-Resource Smart Farming Platforms: Problem Definition, Smallholder Context Mapping, Connectivity Assessment, Affordability Assessment, Digital Inclusion, Modular Architecture, Offline-First and Low-Bandwidth Design, Local Usability, Advisory Integration, Data Governance, Maintenance Planning, Artifact Evaluation, Farmer Feedback, Iteration, and Claim Boundaries

Framework stage	Core purpose	Required input	Design science function	Potential output	Evaluation or assessment need	Failure risk
Low-resource problem definition	Establish a consequential problem without presupposing a digital solution	Farmer, adviser, institutional, infrastructure, and production evidence	Define problem relevance	Bounded problem statement	Stakeholder validation of importance and scope	Technology not solving first problem or constructing a false problem

Smallholder context mapping	Represent heterogeneous farms, users, livelihoods, and practices	Farm systems, production goals, risk, assets, labour, markets, and local knowledge	Define contextual variables and variation points	Smallholder context profile	Assess representation across intended user groups	Treating smallholders as a uniform category
Stakeholder and user mapping	Identify affected users, intermediaries, maintainers, and decision-makers	Farmers, advisers, cooperatives, extension services, providers, technicians, and institutions	Define roles, relationships, and influence	Stakeholder and responsibility map	Representation and power assessment	Omitting less visible or less powerful groups
Connectivity and infrastructure assessment	Determine which technical dependencies are feasible	Coverage, outages, bandwidth, electricity, devices, hosting, charging, and repair access	Constrain architecture and service assumptions	Infrastructure dependency profile	Measurement under realistic operating conditions	Designing continuous services that do not exist
Affordability assessment	Identify the complete financial and operational burden	Acquisition, data, subscription, training, repair, replacement, and user-time costs	Define cost-related requirements and alternatives	Context-specific total-cost profile	Household, cooperative, provider, or institutional affordability analysis	Hidden transferred underestimates costs
Digital inclusion assessment	Identify groups, capabilities, and circumstances associated with exclusion	Gender, age, disability, literacy, language, income, device control, and social position	Define accessibility and alternative-channel requirements	Inclusion-risk profile	Disaggregated access, task, and participation evaluation	Technical access concealing structural exclusion
Platform requirement definition	Translate contextual evidence into testable functions and safeguards	Problem, context, infrastructure, affordability, inclusion, and governance findings	Specify functional and non-functional requirements	Traceable requirement register	Requirement-to-evidence and requirement-to-test audit	Vague, contradictory or unsupported requirements
Modular architecture design	Separate core services from optional or	Functional dependencies, data flows,	Structure the design artifact	Core modules, optional modules, and	Coupling, integration, replacement, and	Hidden dependencies

	replaceable components	integration needs, and maintenance capacity		interface specification	failure-isolation testing	and nominal modularity
Offline-first and low-bandwidth design	Preserve prioritized functions during disconnection or limited throughput	Task criticality, storage, synchronization, payload, and recovery requirements	Instantiate connectivity-resilience mechanisms	Offline workflows and low-data mode	Interruption, latency, conflict, and recovery testing	Data inconsistency, stale output, or incomplete advice
Local usability and accessibility design	Align interaction with language, literacy, devices, abilities, and work practices	User research, task analysis, accessibility needs, and local terminology	Instantiate farmer- and human-centered interaction	Localized and accessible prototype	Task completion, comprehension, error, burden, and accessibility testing	Superficial simplicity, unvalidated localization
Advisory integration design	Connect platform outputs with qualified interpretation and escalation	Advisory roles, institutional capacity, decision risk, and response pathways	Define hybrid digital–human service logic	Advisory referral, shared-record, and escalation workflow	Handoff, response, accountability, and workload assessment	Unclear responsibility or unmanageable adviser demand
Data governance and privacy review	Define justified collection, access, reuse, protection, and farmer control	Data inventory, purposes, actor roles, permissions, retention, and threats	Specify governance and protection mechanisms	Data-governance and security specification	Consent comprehension, access-control, privacy, security, and accountability assessment	Extractive practices, nominal consent, unauthorized access
Maintenance and support planning	Establish how services remain operational	Skills, parts, funding, update processes, documentation, and support actors	Specify non-artifact implementation requirements	Maintenance, repair, training, and support plan	Uptime, serviceability, response, and capacity assessment	Abandonment after routine failures
Artifact evaluation planning	Match each design claim to appropriate evidence	Requirements, artifact maturity, risks, and intended use	Define staged evaluation	Claim-to-test evaluation protocol	Conceptual, technical, usability, governance, implementation, and field assessment	Premature claims based on inadequate measures
Farmer feedback and iteration	Incorporate user experience	Structured feedback, observed use, errors, non-use,	Operate the iterative design cycle	Revised requirements, interfaces, modules, and	Representativeness and response-quality assessment	Extracting feedback without visibility

No deployment-readiness boundary	Prevent conceptual coherence from being treated as operational readiness	Field performance, reliability, safety, governance, maintenance, and support evidence	Establish final maturity control	Deployment-readiness gap statement	Independent multi-domain readiness assessment	Premature operational
----------------------------------	--	---	----------------------------------	------------------------------------	---	-----------------------

Figure 2 presents the proposed design science framework for low-resource smart farming platforms, integrating problem definition, smallholder context mapping, connectivity assessment, affordability assessment, platform requirements, modular architecture, local usability, governance, evaluation, farmer feedback, and framework updating.

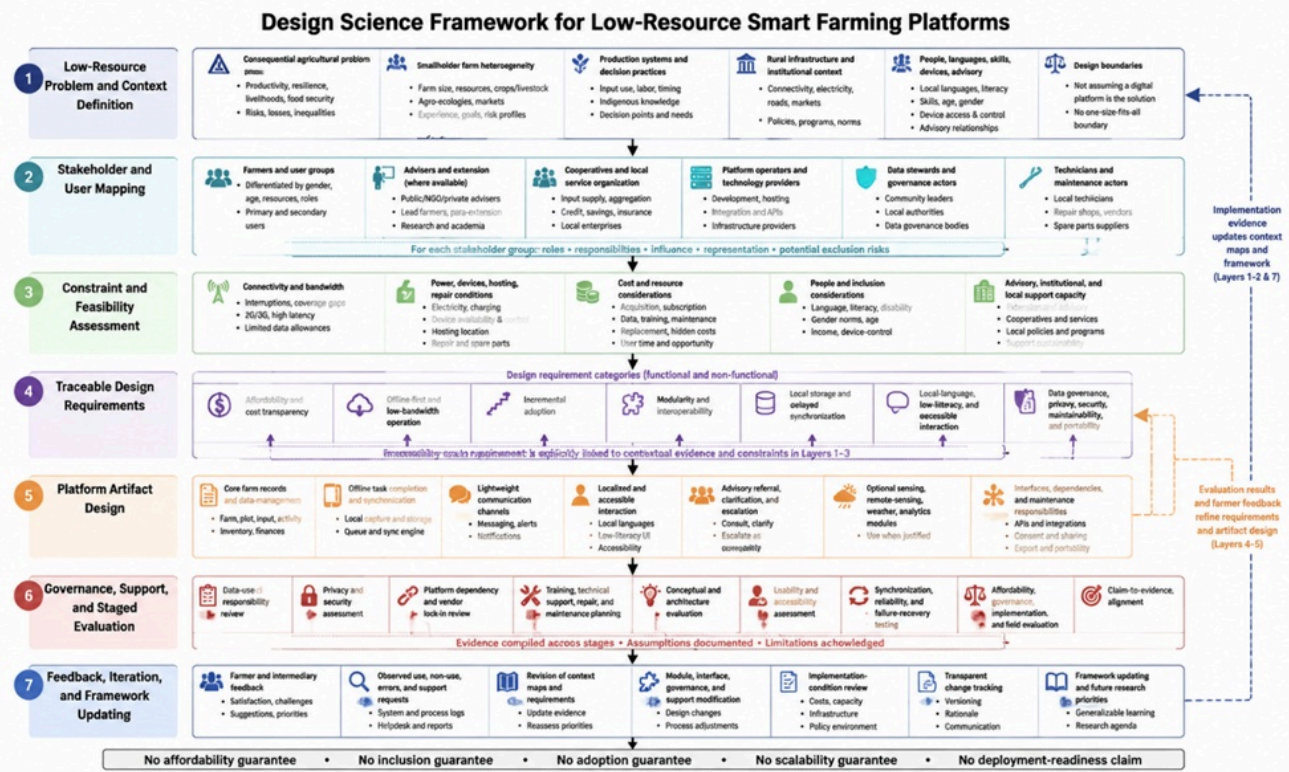


Figure 2. Design Science Framework for Low-Resource Smart Farming Platforms

Alt-text description:

A conceptual design science framework showing stages from low-resource problem definition and smallholder context mapping through connectivity assessment, affordability and inclusion assessment, modular and offline-first platform design, usability, advisory integration, data governance, evaluation, farmer feedback, and framework updating.

Practical Implications and Future Research

For farmers, advisers, extension services, and cooperatives, the framework provides a structure for identifying which platform functions correspond to actual problems and what forms of support are necessary before implementation is considered. Farmer-centered evaluation should examine whether intended users can understand and complete core tasks, question platform outputs, manage permissions, recover from errors, and obtain assistance. Extension and cooperative actors should be evaluated as potential participants in hybrid advisory, training, maintenance, or shared-

access arrangements only where their capacity, legitimacy, and resources support those roles. Participatory processes can help reveal local priorities and exclusion risks, but future studies should examine whose views are represented, how disagreements are handled, and whether farmer input results in traceable design changes [15].

For platform developers, technology providers, and data stewards, the framework emphasizes low-bandwidth usability testing, offline recovery, modularity verification, cost transparency, data portability, maintenance planning, and understandable governance controls. Future research should compare core and optional module configurations without assuming that more integration produces greater utility. It should investigate synchronization failures, device sharing, local-language comprehension, low-literacy interaction, accessibility, data-cost burden, repair pathways, and continuing support. Longitudinal research is particularly important because initial usability or adoption intention may not predict sustained use, adaptation, discontinuation, or dependence on intermediaries. Mobile availability should not be treated as a sufficient indicator of reach, inclusion, or service value [19].

For researchers, public institutions, and policymakers, the framework identifies evidence needs rather than prescribing a universal intervention, procurement model, subsidy, platform, or governance arrangement. Comparative studies are needed to examine affordability across different cost-bearing arrangements, interoperability across modules and providers, data-control usability, maintenance capacity, advisory workloads, and distributional effects. Research should also examine whether platform ecosystems create switching barriers, surveillance risks, asymmetric benefit sharing, or new forms of exclusion. These implications are conceptual research directions rather than agronomic advice, legal advice, policy instructions, donor-programme guidance, platform-selection recommendations, development-project plans, or deployment procedures. Local validation and accountable governance remain necessary before any context-specific operational decision is made [27].

Conclusion

This article developed a conceptual design science framework for low-resource smart farming platforms that connects smallholder context, connectivity and infrastructure constraints, affordability, modularity, interoperability, local usability, advisory integration, data governance, maintenance, evaluation, farmer feedback, and framework updating. Its central proposition is that low-resource agricultural platforms should be designed from contextual requirements rather than transferred directly from high-resource digital agriculture models. Offline-first operation, low-bandwidth communication, incremental adoption, accessible interfaces, modular architecture, farmer data control, and local support may contribute to more appropriate artifact designs, but none independently establishes affordability, inclusion, adoption, scalability, sustainability, or deployment readiness. These outcomes require staged and locally grounded assessment of utility, cost, accessibility, reliability, repairability, governance, institutional capacity, and sustained implementation. The framework therefore offers bounded design knowledge and an evaluation structure rather than a universal platform standard or operational prescription.

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 18 Sep 2025

Revised: 21 Oct 2025

Accepted: 06 Dec 2025

Published online: 25 May 2026

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

Klerkx L, Jakku E, Labarthe P. A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. *NJAS Wageningen J Life Sci.* 2019;90-91:100315.

<https://doi.org/10.1016/j.njas.2019.100315>.

Fielke S, Taylor B, Jakku E. Digitalisation of agricultural knowledge and advice networks: A state-of-the-art review. *Agric Syst.* 2020;180:102763.

<https://doi.org/10.1016/j.agsy.2019.102763>.

Rijswijk K, Klerkx L, Bacco M, Bartolini F, Bulten E, Debruyne L, et al. Digital transformation of agriculture and rural areas: A socio-cyber-physical system framework to support responsabilisation. *J Rural Stud.* 2021;85:79-90.

<https://doi.org/10.1016/j.jrurstud.2021.05.003>.

Rose DC, Wheeler R, Winter M, Lobley M, Chivers CA. Agriculture 4.0: Making it work for people, production, and the planet. *Land Use Policy.* 2021;100:104933.

<https://doi.org/10.1016/j.landusepol.2020.104933>.

Gumbi N, Gumbi L, Twinomurizi H. Towards sustainable digital agriculture for smallholder farmers: A systematic literature review. *Sustainability (Basel).* 2023;15(16):12530.

<https://doi.org/10.3390/su151612530>.

Mizik T. How can precision farming work on a small scale? A systematic literature review. *Precis Agric.* 2023;24(1):384-406.

<https://doi.org/10.1007/s11119-022-09934-y>.

Bayih AZ, Morales J, Assabie Y, de By RA. Utilization of Internet of Things and wireless sensor networks for sustainable smallholder agriculture. *Sensors (Basel).* 2022;22(9):3273.

<https://doi.org/10.3390/s22093273>.

Yuan Y, Sun Y. Practices, challenges, and future of digital transformation in smallholder agriculture: Insights from a literature review. *Agriculture (Basel).* 2024;14(12):2193.

<https://doi.org/10.3390/agriculture14122193>.

Baumüller H. The little we know: An exploratory literature review on the utility of mobile phone-enabled services for smallholder farmers. *J Int Dev.* 2018;30(1):134-54.

<https://doi.org/10.1002/jid.3314>.

Smidt HJ, Jokonya O. Factors affecting digital technology adoption by small-scale farmers in agriculture value chains (AVCs) in South Africa. *Inf Technol Dev.* 2022;28(3):558-84.
<https://doi.org/10.1080/02681102.2021.1975256>.

Steinke J, van Etten J, Müller A, Ortiz-Crespo B, van de Gevel J, Silvestri S, et al. Tapping the full potential of the digital revolution for agricultural extension: An emerging innovation agenda. *Int J Agric Sustain.* 2021;19(5-6):549-65.
<https://doi.org/10.1080/14735903.2020.1738754>.

Choruma DJ, Dirwai TL, Mutenje MJ, Mustafa M, Chimonyo VGP, Jacobs-Mata I, et al. Digitalisation in agriculture: A scoping review of technologies in practice, challenges, and opportunities for smallholder farmers in sub-Saharan Africa. *J Agric Food Res.* 2024;18:101286.
<https://doi.org/10.1016/j.jafr.2024.101286>.

Gregor S, Chandra Kruse L, Seidel S. Research perspectives: The anatomy of a design principle. *J Assoc Inf Syst.* 2020;21(6):1622-52.
<https://doi.org/10.17705/1jais.00649>.

Prost L. Revitalizing agricultural sciences with design sciences. *Agric Syst.* 2021;193:103225.
<https://doi.org/10.1016/j.agsy.2021.103225>.

Steinke J, Ortiz-Crespo B, van Etten J, Müller A. Participatory design of digital innovation in agricultural research-for-development: Insights from practice. *Agric Syst.* 2022;195:103313.
<https://doi.org/10.1016/j.agsy.2021.103313>.

Eastwood CR, Turner FJ, Romera AJ. Farmer-centred design: An affordances-based framework for identifying processes that facilitate farmers as co-designers in addressing complex agricultural challenges. *Agric Syst.* 2022;195:103314.
<https://doi.org/10.1016/j.agsy.2021.103314>.

Kenny U, Regan Á, Hearne D, O'Meara C. Empathising, defining and ideating with the farming community to develop a geotagged photo app for smart devices: A design thinking approach. *Agric Syst.* 2021;194:103248.
<https://doi.org/10.1016/j.agsy.2021.103248>.

Lamsal RR, Karthikeyan P, Otero P, Ariza A. Design and implementation of Internet of Things (IoT) platform targeted for smallholder farmers: From Nepal perspective. *Agriculture (Basel).* 2023;13(10):1900.
<https://doi.org/10.3390/agriculture13101900>.

Muromba P, Keeni M, Fuyuki K. A systematic review of mobile agricultural service applications for smallholder farmers in sub-Saharan Africa: Perspectives from the technology acceptance model. *Agric Food Secur.* 2025;14:34.
<https://doi.org/10.1186/s40066-025-00563-y>.

Wolfert S, Ge L, Verdouw C, Bogaardt MJ. Big data in smart farming—a review. *Agric Syst.* 2017;153:69-80.
<https://doi.org/10.1016/j.agsy.2017.01.023>.

Mushi GE, Di Marzo Serugendo G, Burgi PY. Data management system for sustainable agriculture among smallholder farmers in Tanzania: Research-in-progress. *Inf Technol Dev.* 2023;29(4):558-81.
<https://doi.org/10.1080/02681102.2023.2215528>.

Wiseman L, Sanderson J, Zhang A, Jakku E. Farmers and their data: An examination of farmers' reluctance to share their data through the lens of the laws impacting smart farming. *NJAS Wageningen J Life Sci.* 2019;90-91:100301.
<https://doi.org/10.1016/j.njas.2019.04.007>.

Jakku E, Taylor B, Fleming A, Mason C, Fielke S, Sounness C, et al. "If they don't tell us what they do with it, why would we trust them?" Trust, transparency and benefit-sharing in smart farming. *NJAS Wageningen J Life Sci.* 2019;90-91:100285.
<https://doi.org/10.1016/j.njas.2018.11.002>.

Wolfert S, Verdouw C, van Wassenhaer L, Dolfsma W, Klerkx L. Digital innovation ecosystems in agri-food: Design principles and organizational framework. *Agric Syst.* 2023;204:103558.

<https://doi.org/10.1016/j.agsy.2022.103558>.

Krisnawijaya NNK, Tekinerdogan B, Catal C, van der Tol R. Reference architecture design for developing data management systems in smart farming. *Ecol Inform.* 2024;81:102613.

<https://doi.org/10.1016/j.ecoinf.2024.102613>.

Njuguna E, Daum T, Birner R, Mburu J. Silicon Savannah and smallholder farming: How can digitalization contribute to sustainable agricultural transformation in Africa? *Agric Syst.* 2025;222:104180.

<https://doi.org/10.1016/j.agsy.2024.104180>.

Sarku R, Ayanga M. Is the right going wrong? Analysing digital platformization, data extractivism and surveillance practices in smallholder farming in Ghana. *Inf Technol Dev.* 2025;31(4):1039-65.

<https://doi.org/10.1080/02681102.2024.2447596>.