

Permissions

Securing Data and Scaling SmartSuite Processes



Executive Summary

The Permissions phase ensures that information and capabilities are accessible to the right people, while protecting sensitive information and supporting compliance requirements. Without deliberate access design, organizations risk exposing confidential data, creating bottlenecks, or undermining trust in the system. Improper permissions can also lead to errors, conflicting updates, or inconsistent enforcement of business rules, issues that are difficult and costly to correct once the solution is live.

When approached intentionally, this phase provides significant benefits. A well-structured permissions model improves security, supports role-based workflows, and reduces operational friction. By clearly defining access at the solution, table, record, and field levels, teams can balance usability with governance, ensure appropriate segregation of duties, and scale access controls as teams grow or organizational requirements change.



This white paper focuses on designing effective, scalable, and maintainable access controls in SmartSuite. It outlines best practices for role definition, least-privilege access, dynamic and consistent rule application, and permissions governance. The guidance also demonstrates how these principles can be applied to a sample process, providing a tangible example of permission design in practice and highlighting how thoughtful planning prevents downstream issues.

The intended audience includes solution owners, administrators, and implementation teams responsible for controlling access while maintaining productivity. For these stakeholders, a well-designed permissions framework ensures that data is both secure and usable, supports compliance requirements, reduces risk, and enables teams to focus on executing their processes efficiently without worrying about unintended data exposure or access errors.

Why the Permissions Phase Matters

The Permissions phase is focused on ensuring that users have the right access to the right information at the right time. It encompasses defining roles, configuring access at the solution, table, record, and field levels, and implementing controls that enforce segregation of duties. By aligning access with responsibilities, organizations can protect sensitive data, comply with regulations, and maintain operational efficiency without introducing unnecessary friction.

Neglecting permissions best practices can have immediate and lasting consequences. Overly broad access exposes confidential information and increases risk, while overly restrictive access slows workflow, frustrates users, and drives workarounds outside of the system. Correcting these issues after deployment is disruptive, costly, and can erode trust in both the solution and the processes it supports.

The importance of this phase lies in establishing a secure, scalable foundation that supports usability and governance simultaneously. Thoughtful permission design ensures that users can efficiently complete their work, information flows only to those who need it, and compliance obligations are met. By enforcing the Principle of Least Privilege, teams can reduce risk while maintaining transparency and accountability across processes.

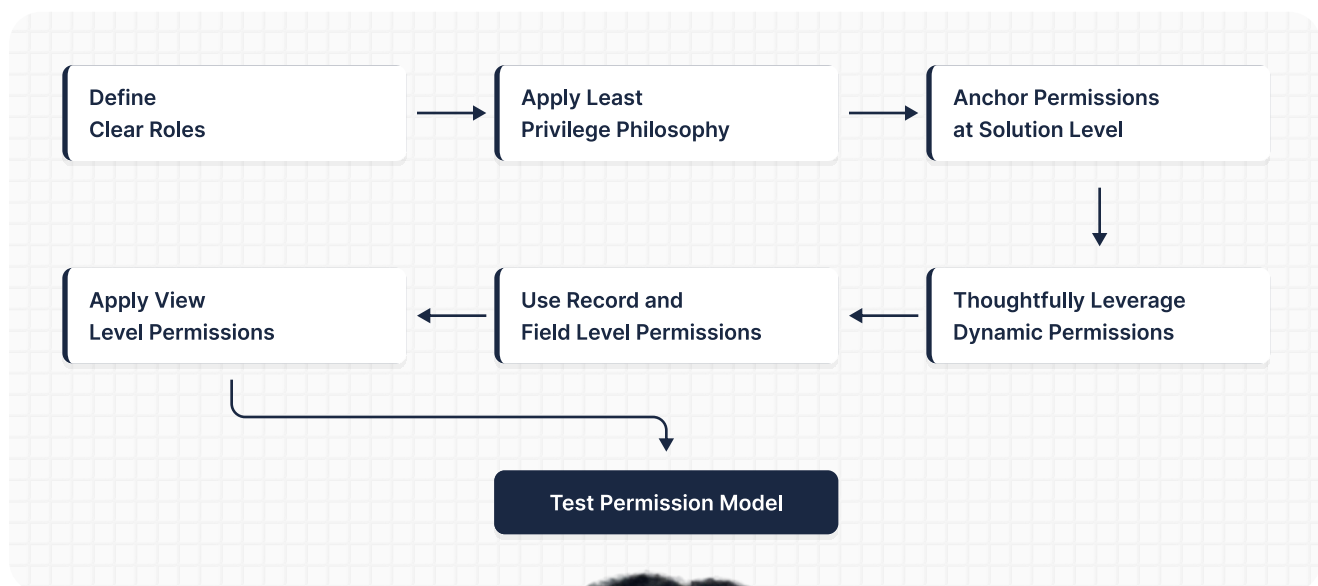
In addition, the Permissions phase provides a framework for ongoing governance and monitoring. Defining ownership for access rules, implementing change tracking, and periodically reviewing permissions helps maintain system integrity as teams grow and processes evolve. For solution owners, administrators, and implementation teams, investing in this phase ensures a balance between productivity, security, and compliance that sets the stage for long-term adoption and trust.



Permission Phase Best Practices

The Permissions phase requires deliberate, structured decision-making to ensure that access aligns with roles, responsibilities, and organizational policies. Applying best practices helps teams balance usability with data protection, enforce the Principle of Least Privilege, and establish a scalable access model that supports current and future needs.

This section outlines actionable recommendations for defining roles, configuring solution- and record-level access, managing field visibility, implementing governance guardrails, and monitoring changes, all with the goal of creating a secure, consistent, and maintainable environment that minimizes risk while keeping workflows efficient.



PHASE 1**Start with Clearly Defined Roles, Not Individual Users**

Establish roles based on responsibilities and outcomes, then map each role to the access it truly needs. Designing permissions around roles allows access models to scale cleanly as teams grow or responsibilities shift.

PHASE 2**Apply a Least Privilege Philosophy by Default**

Grant the minimum access required for users to perform their tasks, expanding permissions only when justified by process needs. This approach reduces risk, simplifies audits, and minimizes unintended data exposure.

PHASE 3**Anchor Permissions at the Solution Level and Narrow Only When Needed**

Use SmartSuite's solution-level permissions as the primary control point, then selectively refine access at the table, record, or field level. This layered approach keeps the model understandable while preserving flexibility.

PHASE 4**Use Dynamic Permissions Sparingly and With Clear Logic**

Favor simple, unified rules that produce consistent outcomes across records. Dynamic permissions should be easy to explain and validate; overly complex rule sets increase maintenance burden and raise the likelihood of access errors.

PHASE 5

Leverage Record- and Field-Level Controls for Sensitive Data

Apply finer-grained permissions only where data sensitivity or segregation of duties requires it. Field-level visibility controls are especially effective for protecting confidential attributes without duplicating tables or workflows.

PHASE 6

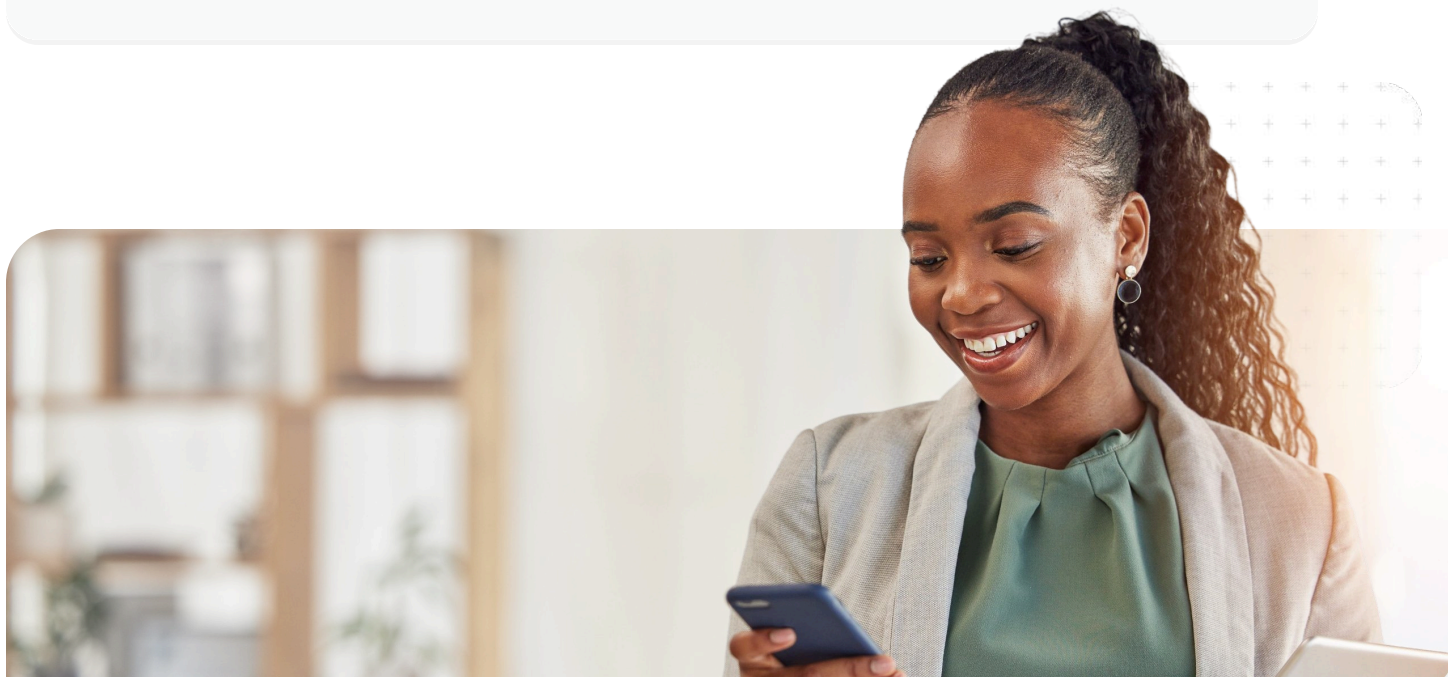
Incorporate View-Level Permissions to Guide User Interaction

Use views to shape what users see and how they engage with records. Restricting access through role-appropriate views reinforces permissions, improves usability, and reduces the chance of unintended actions.

PHASE 7

Test Permissions Across Realistic User Scenarios

Validate access using representative roles and edge cases before go-live. Permission testing should confirm not only what users can see, but also what actions they can and cannot perform at each stage of the workflow.



Understanding SmartSuite's Permission Layers

SmartSuite uses a layered permission model that allows access to be defined broadly and then refined only where necessary. Each layer serves a distinct purpose, enabling solution owners to balance simplicity, security, and scalability.

Rather than relying on a single control point, permissions work together to ensure users see and act on exactly what they should, no more, no less.

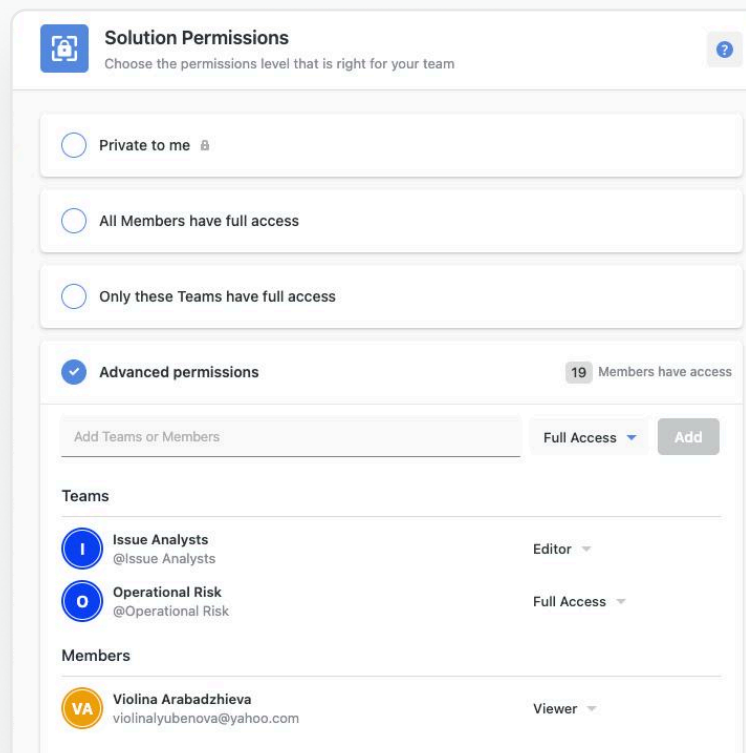
Solution-Level Permissions: Establishing the Access Baseline

Solution permissions define who can access a solution and at what level. This layer establishes the foundational access model that applies across all tables and workflows by default.

At the solution level, owners can:

- Control overall visibility of the solution
- Assign role-based access aligned to responsibilities
- Limit access to specific teams rather than individuals

Solution-level permissions should act as the primary anchor point for access design. In most cases, this layer alone governs access for the majority of users, allowing the solution to scale cleanly as teams evolve.



Solution Permissions
Choose the permissions level that is right for your team

☐ Private to me

☐ All Members have full access

☐ Only these Teams have full access

☒ **Advanced permissions** 19 Members have access

Add Teams or Members Full Access Add

Teams

 Issue Analysts @Issue Analysts	Editor
 Operational Risk @Operational Risk	Full Access

Members

 Violina Arabadzhieva violinalyubenova@yahoo.com	Viewer
---	--------

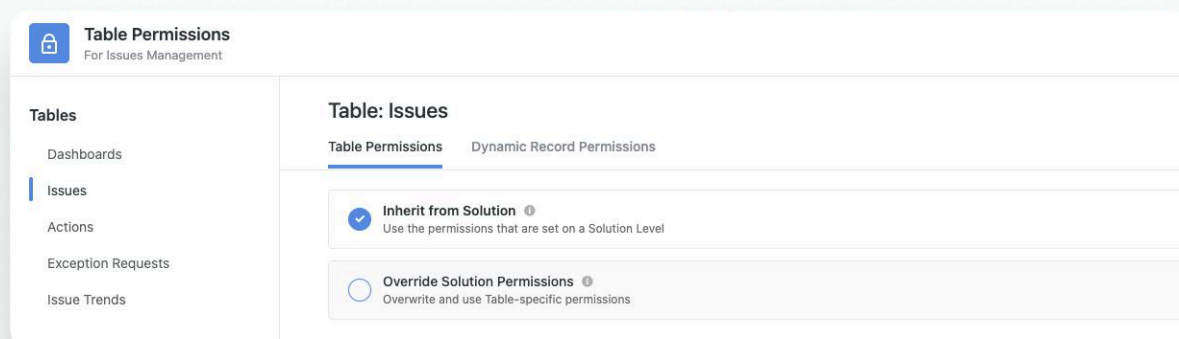
Table-Level Permissions: Differentiating Access by Data Domain

Table permissions allow solution owners to refine access for specific datasets within a solution. This is particularly important when a solution contains tables with different sensitivity levels, ownership models, or operational purposes.

Table-level controls enable:

- Restricting access to sensitive or regulated datasets
- Granting broader access to intake or reference tables
- Supporting segregation of duties across related processes

These permissions narrow solution-level access rather than replacing it, preserving a clear and predictable permission structure.



Field-Level Permissions: Protecting Sensitive Attributes

Field-level permissions provide fine-grained control over individual data elements within a record. This allows sensitive information to be protected without fragmenting tables or workflows.

Common use cases include:

- Restricting financial, personnel, or compliance-related fields
- Allowing broad record visibility while hiding confidential attributes
- Supporting oversight and review roles without overexposure

Field-level controls are most effective when applied selectively, reinforcing least-privilege access without adding unnecessary complexity.



Dynamic Record Permissions: Context-Aware Access at Scale

Dynamic record permissions are the most powerful and flexible layer in SmartSuite's permission model. They enable access to be evaluated at the individual record level, based on business context rather than static role assignment.

This layer allows access to change automatically as records:

- Are created or assigned
- Move through workflow stages
- Meet defined conditions or criteria

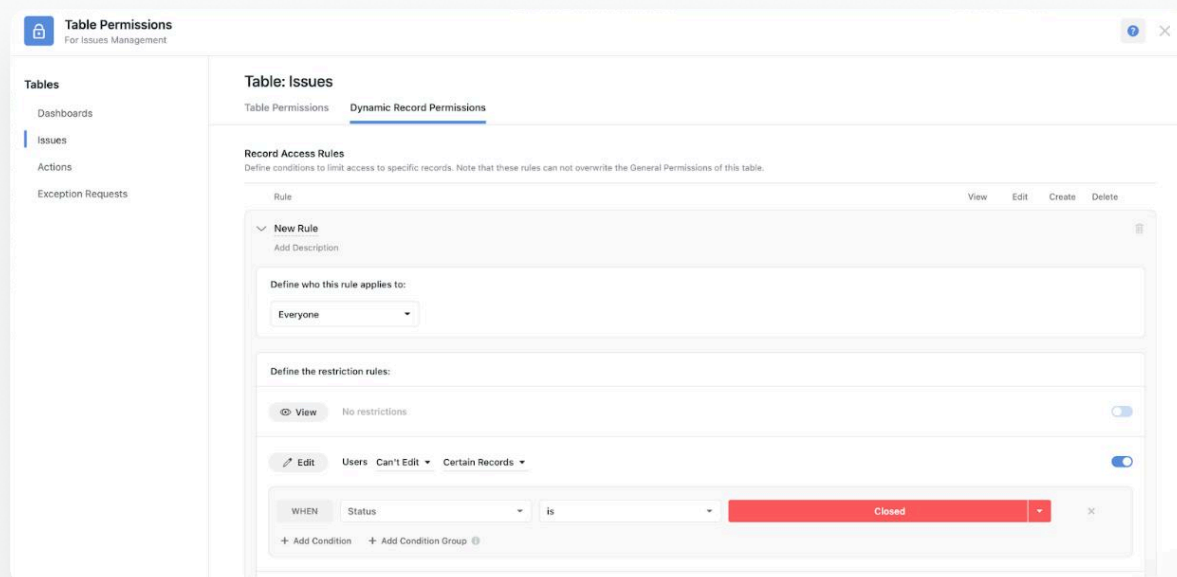
With dynamic record permissions, solution owners can:

- Ensure users see only records relevant to their responsibilities
- Enforce segregation of duties within shared workflows
- Support approvals, escalations, and reviews without duplicating data
- Maintain a single source of truth while serving multiple audiences securely

Because dynamic record permissions operate within the boundaries of solution- and table-level access, they refine visibility without introducing risk. When multiple rules apply, the most restrictive outcome always prevails, ensuring predictable and defensible access behavior.

Why this matters:

Dynamic record permissions transform SmartSuite from a static, role-based access model into a living permission system that mirrors real operational processes. This capability dramatically expands what can be built in a single solution while reducing administrative overhead and long-term maintenance.



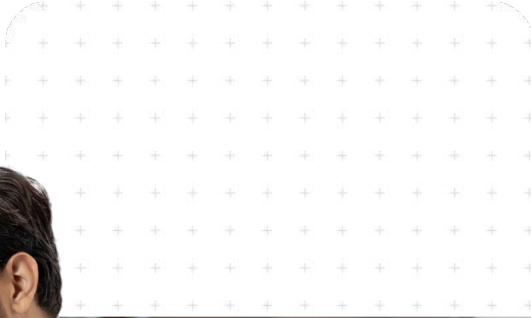
View (Folder) Permissions: Guiding User Interaction

Folder and view permissions shape how users experience and interact with data, even when underlying access remains unchanged.

These permissions are commonly used to:

- Present role-specific perspectives on shared data
- Hide administrative or audit views from contributors
- Reinforce process flow and reduce unintended actions

While view permissions do not change underlying data access, they play a critical role in usability, clarity, and adoption.



**Solution Permissions**
Choose the permissions level that is right for your team

☐ Private to me

☒**Field Permissions**
Choose the permissions level that is right for your team

Who can View this Field?

 Everyone

Who can Edit this Field?

 Selected Members or Groups

Common Permission Pitfalls

Even with a well-defined model for access, permissions can fail if common mistakes are made during design and implementation. These pitfalls often lead to overexposed data, operational risk, user confusion, or increased maintenance overhead. Understanding the typical challenges and knowing how to address them proactively helps ensure that access controls remain secure, consistent, and scalable as teams and processes evolve. The following list highlights frequent permission missteps and practical ways to avoid them.

Designing Permissions Around Individuals Instead of Roles

Assigning access based on named users rather than clearly defined roles leads to inconsistent access, manual upkeep, and increased risk as teams change.

Permissions should be role-based and scalable by design.

How to Avoid It:

Base all access on clearly defined roles rather than individual users, ensuring permissions scale as teams change.

Over-Permissive Access “To Keep Things Moving”

Granting broad access early in an implementation often becomes permanent, increasing data exposure and compliance risk. Least-privilege access should be the default, with temporary exceptions documented and reviewed.

How to Avoid It:

Default to least-privilege access, documenting any temporary exceptions and reviewing them regularly.

Inconsistent Permission Models Across Tables

Applying different access patterns to similar tables confuses users and complicates governance. Permission structures should be consistent wherever possible to support predictability and easier maintenance.

How to Avoid It:

Apply uniform access patterns wherever possible to make permissions predictable and easier to maintain.

Ignoring Record-Level and Field-Level Sensitivity

Treating all records or fields the same can expose sensitive data unnecessarily. SmartSuite's record- and field-level controls should be used to protect confidential or restricted information.

How to Avoid It:

Use SmartSuite's record- and field-level controls to protect confidential or restricted information appropriately.

Failing to Enforce Segregation of Duties

Allowing the same role to create, approve, and close records introduces operational and audit risk. Segregation-of-duties patterns should be intentionally designed and tested.

How to Avoid It:

Intentionally design and test role separation so that critical actions, such as create, approve, and close, are distributed across multiple roles.

Uncontrolled External or Guest Access

Providing guest access without clearly defining scope, expiration, and ownership can create blind spots. External access should be limited, monitored, and periodically reviewed.

How to Avoid It:

Limit external access, define its scope and expiration, assign clear ownership, and monitor usage regularly.

No Formal Access Review or Certification Process

Permissions that are never reviewed drift over time as roles and responsibilities change. Regular access reviews help ensure permissions remain appropriate and defensible.

How to Avoid It:

Conduct periodic access reviews to confirm permissions remain aligned with roles, responsibilities, and organizational policies.

Insufficient Permission Testing Before Go-Live

Testing permissions only as administrators misses real-world scenarios. Permission validation should include role-based testing using realistic user personas.

How to Avoid It:

Validate permissions using realistic user personas to test workflows from the perspective of end users, not just administrators.

High Level Permissions Checklist

The checklist below supports permission design and validation, including role mapping, segregation-of-duties checks, and access review procedures.

Checklist Item	Description	Benefits
Define Role-Based Access Model	Establish roles based on responsibilities and outcomes rather than individual users.	Scalable access, reduced maintenance
Apply Least Privilege by Default	Grant only the access required for each role, expanding only when justified.	Reduced risk and data exposure
Anchor Permissions at the Solution Level	Use solution-level permissions as the primary control point, refining only where necessary.	Consistency and clarity
Differentiate Access by Data Sensitivity	Restrict access at the table or field level for sensitive or regulated data.	Targeted protection without fragmentation
Use Dynamic Record Permissions Where Context Matters	Adjust access automatically based on record ownership, status, or workflow stage.	Context-aware security at scale
Enforce Segregation of Duties	Ensure no role controls conflicting steps within the same process.	Audit readiness and operational integrity
Guide Interaction with Role-Based Views	Use views to shape how users see and work with data.	Improved usability and fewer errors
Control and Review External Access	Clearly define scope, ownership, and review cycles for guest or external users.	Safer collaboration
Validate Permissions with Realistic Scenarios	Test access using representative user roles before go-live.	Fewer post-launch corrections

Permissions Use Case Example: Issue & Action Tracking

In the Issue & Action Tracking solution, permissions ensured contributors, action owners, and reviewers each had appropriate access without exposing sensitive escalation data.

Permission Configuration	Benefit
Restrict general users to their assigned issues – Users can only view or update records assigned to them or within their area of responsibility.	Ensures employees focus only on relevant issues, reduces clutter, and prevents exposure of unrelated sensitive data.
Assurance functions see all issues within their area – Second- and third-line assurance teams can access all records in their scope but not outside their area.	Allows assurance teams to monitor and manage their portfolio effectively while maintaining segregation of duties and limiting unnecessary exposure.
Field- and record-level sensitivity restrictions – Certain fields, like HR or Legal notes, are visible only to users with the appropriate role.	Protects confidential information while ensuring sensitive data is available only to those who need it, maintaining compliance and trust.
Locking records after completion or validation – Once an issue is closed, validated, or approved, general users cannot modify the record; only authorized roles can make adjustments.	Preserves data integrity, prevents accidental changes, and provides a clear audit trail of completed issues.
Role-based assignment and review permissions – Users in review or oversight roles can assign issues, escalate, or approve actions without editing unrelated content.	Separates responsibilities for operational execution and oversight, supporting accountability, governance, and compliance.

Conclusion: Securing Access, Enabling Trust

Properly designed permissions are the backbone of a reliable and scalable SmartSuite solution. By defining roles thoughtfully, applying least-privilege principles, and implementing record- and field-level controls, organizations can ensure users have the access they need without exposing sensitive data or creating operational risk.

A disciplined approach to permissions supports efficiency, accountability, and compliance. Regular reviews, clear ownership, and consistent enforcement prevent drift over time and keep workflows predictable. When permissions are configured intentionally, teams can confidently rely on SmartSuite to guide execution, protect critical information, and maintain trust in both the data and the system itself.

