A REFERENCE ARCHITECTURE FOR OPERATIONS LEADERS

When information doesn't flow, work doesn't either.

The same information gets typed into a spreadsheet, then a form, then a system, then a report — and somewhere in that chain, quality, time, and trust all leak out.

Forms & portals — Integration layer — Core platform — Systems of record

A conceptual reference architecture for complex project lifecycles →

THE HIDDEN COST

Every handoff is a chance to lose information

In most complex project environments, the same facts travel through five or six different formats before they ever reach a system of record.

Email threads

Decisions buried in inboxes, invisible to the next person in the chain

Spreadsheets

Parallel “source of truth” files that quietly drift out of sync

PDFs & forms

Static documents that have to be manually re-keyed elsewhere

Disconnected systems

Each platform holds a partial, slightly different version of reality

Enter once.

Reuse everywhere.

When the same record feeds every downstream process, accuracy stops being a matter of discipline — it becomes a property of the system.

REDESIGNING THE WORKFLOW

One lifecycle, not eight handoffs

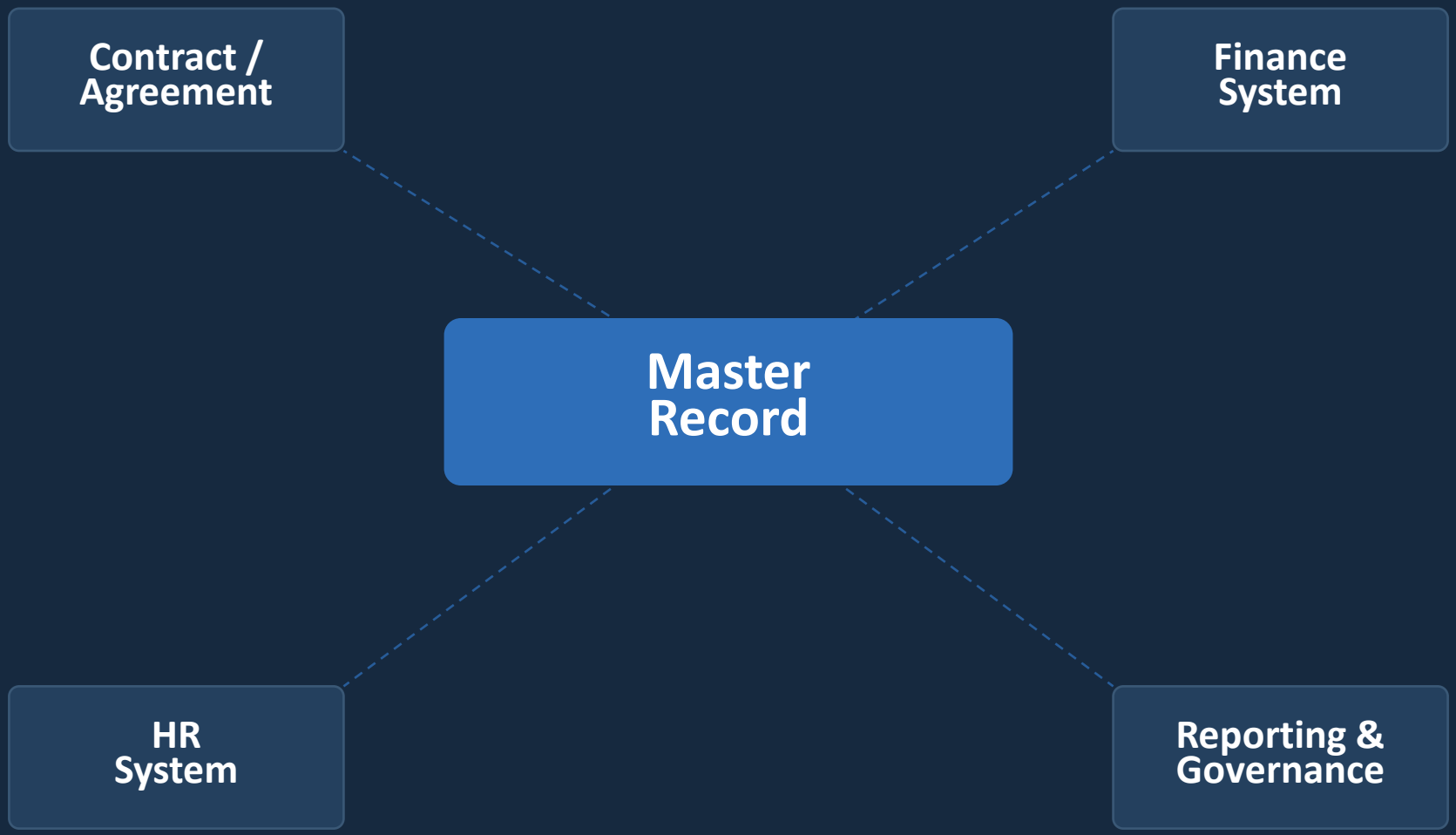
A generic project lifecycle, redrawn so each stage reads from and writes to a single record — instead of re-asking questions already answered.

1. Intake / Request
2. Decision
3. Agreement Received
4. Review
5. Setup Request
6. **Record Created** — MASTER RECORD
7. Activation

THE MASTER RECORD

One authoritative record. Many consumers.

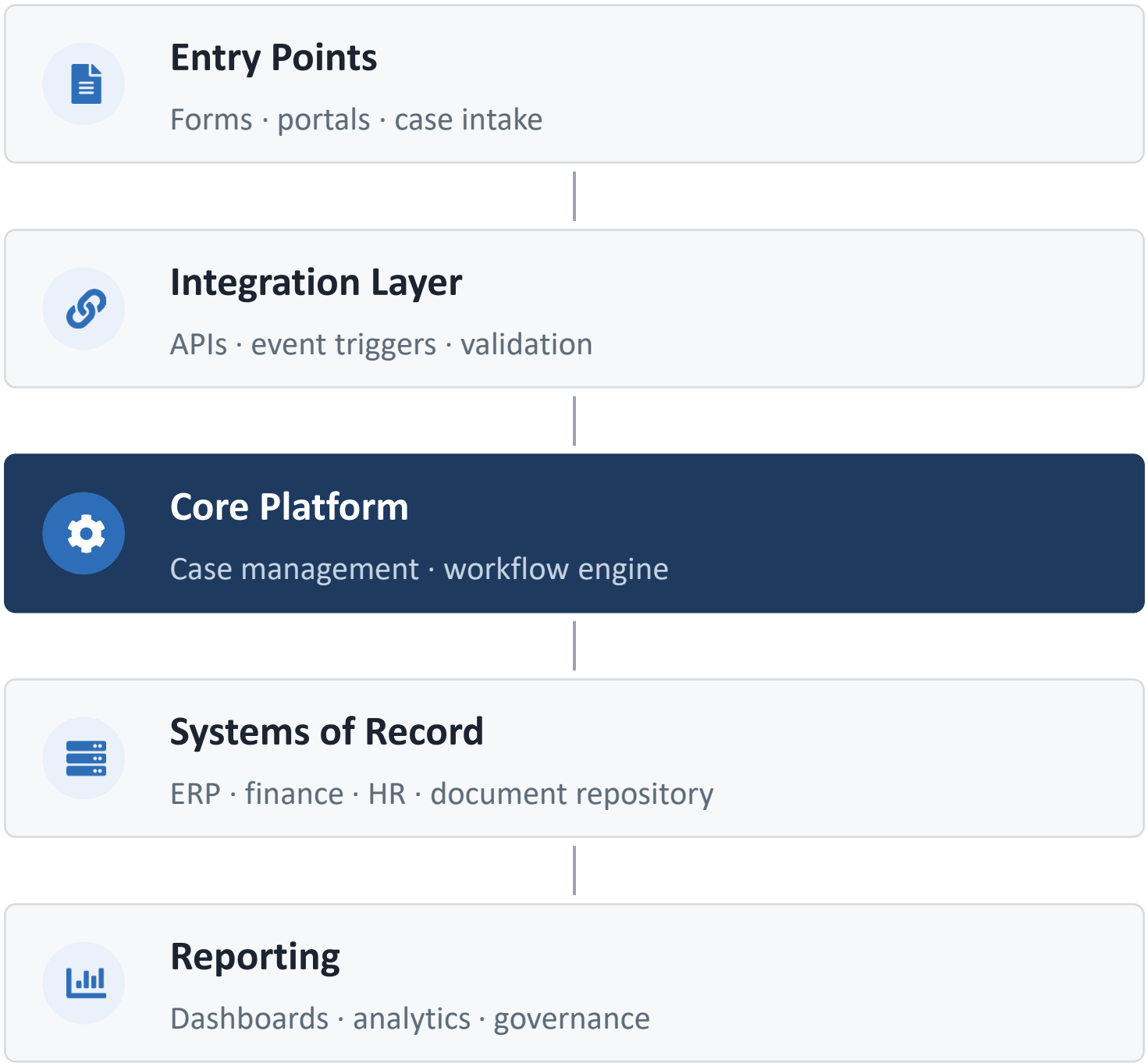
Defined once, owned clearly, validated at the source — every downstream process reads from it rather than re-collecting the same facts.



Single source of truth · clear data ownership · validation at the point of entry

SYSTEM ARCHITECTURE

A vendor-neutral reference layer model



AUTOMATION OPPORTUNITIES

Replace manual approvals with event-driven workflows

Automation isn't about removing people — it's about reserving human judgement for the decisions that actually need it.

Orchestration

Route work automatically based on type, value, or risk — not on who happens to see the email first.

Validation

Catch missing or inconsistent data at the point of entry, before it ever reaches a downstream system.

Exception Handling

Let the routine path run untouched; surface only the cases that genuinely need a human decision.

Governance

Keep approval logic configurable and auditable, so policy changes don't require rebuilding the process.

OPERATING MODEL SHIFT

From document-driven to data-driven

BEFORE

Document-Driven

- Data re-entered at every stage
- Status tracked in spreadsheets
- Approvals routed by email
- Errors found after the fact
- Reporting built manually, after delays

AFTER

Automation-First

- Data captured once, reused everywhere
- Status updates automatically, in real time
- Approvals routed by policy, not inbox order
- Errors caught at the point of entry
- Reporting available continuously, by design

DESIGN PRINCIPLES

Five principles worth designing around

Single source of truth

One authoritative record per case, not five partial versions.

One-time data entry

Information is captured once and reused, never retyped.

Event-driven automation

Work moves forward automatically when conditions are met.

Configurable governance

Policy and approval logic live in configuration, not in code or memory.

Judgement where it matters

Human review reserved for genuine exceptions, not routine steps.

CLOSING THOUGHT

We often try to automate tasks.

Maybe we should start by redesigning how information flows.

If this resonates with how your organisation handles complex project lifecycles — contracts, grants, claims, cases, or onboarding — I'd love to hear how you're approaching it.