



Der große Leitfaden für InterSystems IRIS

Datenhaltung, Integration, Entwicklung und Betrieb
– so bleibt die IT-Landschaft langfristig beherrschbar,
anpassungsfähig und wirtschaftlich tragfähig

Inhalt

Einleitung

Warum IT-Architektur eine Zukunftsfrage ist

4

Einfach mehr machen: Die Anforderungen wachsen, nicht zuletzt durch Automatisierung und KI. Der strategische Wert einer Plattform entscheidet sich daher in ihrer Fähigkeit, strukturelle Komplexität zu reduzieren und Veränderung beherrschbar zu machen.

Kapitel 1

Architektonische Herausforderungen und Entscheidungskriterien

5

Von der Analyse zur Entscheidung: Woran sich erkennen lässt, wann ein Technologie-Stack zum Risiko wird und wie sich Architektur-Entscheidungen wirtschaftlich und organisatorisch bewerten lassen.

Kapitel 2

Datenarchitektur als Kernentscheidung

10

Von der Datenverwaltung zur Datennutzung: Warum sich die Tragfähigkeit einer Systemlandschaft daran entscheidet, wie Daten organisiert, verbunden und nutzbar gemacht werden.

Kapitel 3

Entwicklung, Umsetzung und risikoarmer Einstieg

16

Turbo für Time-to-Value: Wie die Architektur zum Geschwindigkeitsvorteil wird - und warum Demo-Formate mehr sein müssen als ein bloßer Proof of Concept.

Kapitel 4

Performance, Skalierung und Stabilität

20

Heute beginnen, an morgen denken: So finden Sie einen technischen Pfad, der auch unter veränderten Rahmenbedingungen Bestand hat – und Performance, Skalierbarkeit und Stabilität vereint.

Kapitel 5

Interoperabilität und Legacy-Strategie

26

Das Rad drehen, nicht das Rad neu erfinden: IT-Landschaften sind historisch gewachsen und heterogen. Erfolgreich ist in diesem Umfeld nur, wer Bestehendes integriert, um Neues zu schaffen.



Executive Summary

Dieser Leitfaden richtet sich an technische und strategische Entscheider – darunter CIOs, CTOs, IT- und Enterprise-Architekten sowie Verantwortliche für Datenmanagement, Integration, Anwendungsentwicklung und IT-Betrieb. Er unterstützt sie dabei, fundiert zu bewerten, ob InterSystems IRIS eine geeignete Grundlage für die Weiterentwicklung ihrer IT-Landschaft darstellt. Dabei wird die Plattform nicht als isoliertes Produkt betrachtet, sondern als strategische Entscheidung mit technischen, organisatorischen und wirtschaftlichen Folgen. Im Mittelpunkt steht die Frage, wie Datenhaltung, Integration, Entwicklung und Betrieb so gestaltet werden können, dass die IT-Landschaft langfristig beherrschbar, anpassungsfähig und wirtschaftlich tragfähig bleibt.

InterSystems IRIS ist insbesondere für Unternehmen relevant, die wachsende Integrationsanforderungen bewältigen müssen und zugleich die Komplexität ihrer IT-Infrastruktur reduzieren möchten. Wenn viele Schnittstellen ineinandergreifen und Daten über verschiedene Kontexte hinweg konsistent verfügbar sein sollen, geraten gewachsene Architekturen schnell unter Druck.

Ein integrierter Plattformansatz kann in solchen Situationen helfen, Übergänge zwischen Systemen zu vereinfachen und die weitere Entwicklung auf ein stabileres Fundament zu stellen. Ob dieser Ansatz sinnvoll ist, hängt jedoch nicht allein von der Breite des Funktionsumfangs ab, sondern vor allem von der konkreten Ausgangslage, den bestehenden Anforderungen und der erwarteten Entwicklung des Unternehmens.

Eine belastbare Entscheidung muss daher über einen reinen Funktionsvergleich hinausgehen. Sie sollte berücksichtigen, wie die bestehende Integrationsarchitektur aufgebaut ist, welcher Aufwand für Entwicklung und Betrieb entsteht und welche organisatorischen Voraussetzungen gegeben sind. Besonders aussagekräftig ist ein Proof of Concept anhand eines realistischen Anwendungsfalls. Er zeigt unter realen Bedingungen, wie gut InterSystems IRIS zur bestehenden IT-Landschaft passt und ob sich der erwartete technische und wirtschaftliche Nutzen tatsächlich realisieren lässt.

Kadir Dindar

Managing Director /
General Manager DACH InterSystems



Warum IT-Architektur eine Zukunftsfrage ist

Der strategische Wert einer Datenplattform offenbart sich in ihrer Fähigkeit, strukturelle Komplexität zu reduzieren und Veränderung beherrschbar zu machen.

Unternehmen stehen heute nicht mehr nur vor der Frage, wie sie Daten besser nutzen, sondern ob ihre bestehende Systemlandschaft den steigenden Anforderungen wirtschaftlich und organisatorisch noch standhält. Integrationsdruck, Datenwachstum, neue Analyseanforderungen, Automatisierung und KI-Initiativen treffen auf historisch gewachsene IT-Landschaften. Der eigentliche Engpass liegt dabei schon lange nicht mehr in einzelnen Systemen, sondern in den Verbindungen zwischen ihnen.

Dieser Leitfaden bewertet InterSystems IRIS als strategische Architekturplattform. Im Mittelpunkt steht die Analyse, wie eine solche Plattform unter realen Betriebsbedingungen dauerhaft tragfähig bleibt, wenn Last, Integrationsdichte sowie Anforderungen an Governance und Steuerbarkeit zunehmen. Genau daran entscheidet sich der strategische Wert einer Plattform: in ihrer Fähigkeit, strukturelle Komplexität zu reduzieren und Veränderung beherrschbar zu machen. Entscheidend ist nicht der isolierte Use Case, sondern die Frage, ob ein architektonischer Rahmen unter realen Bedingungen tragfähig bleibt, vor allem auch bei steigender Last, wachsender Integrationsdichte und zunehmenden Governance-Anforderungen. Genau hier offenbart sich der strategische Wert einer Plattform: in ihrer Fähigkeit, strukturelle Komplexität zu reduzieren und Veränderung beherrschbar zu machen.



1 **Architektonische Herausforderungen und Entscheidungskriterien**

Für Unternehmen lautet die Leitfrage nicht, ob ein zusätzlicher Baustein sinnvoll ist, sondern ob die Struktur der IT-Landschaft mit jedem weiteren Baustein besser oder schwerer steuerbar wird.

Die Qualität einer Architektur zeigt sich selten im stabilen Betrieb einzelner Systeme, sondern unter wachsender Komplexität im Zusammenspiel des Ganzen. Strategisch bedeutsam wird sie meist dann, wenn mit zunehmender Vernetzung auch Anbindungsaufwand, Betriebsaufwand, Steuerungsbedarf und Abhängigkeiten wachsen. Genau an diesem Punkt geht es nicht mehr nur um die Auswahl geeigneter Einzelwerkzeuge, sondern um die Tragfähigkeit der Gesamtarchitektur. Für Unternehmen lautet die Leitfrage daher nicht, ob ein zusätzlicher Baustein lokal sinnvoll ist, sondern ob die Struktur der IT-Landschaft mit jedem weiteren Baustein noch besser oder bereits schwerer steuerbar wird. Die folgenden Abschnitte zeigen, woran sich erkennen lässt, wann ein gewachsener Technologie-Stack zum Risiko wird, nach welchen Kriterien sich Technologie-Stack und Plattform sinnvoll gegeneinander abwägen lassen und wie sich diese Entscheidung wirtschaftlich und organisatorisch bewerten lässt.

1.1 **Wann wird ein Technologie-Stack zu einem strategischen Risiko?**

Technologie-Stacks entstehen in der Praxis selten als Ergebnis einer konsequent verfolgten Zielarchitektur. Meist wachsen sie schrittweise: Ein neues Projekt bringt ein zusätzliches Werkzeug mit sich, eine Integrationsanforderung erzeugt eine weitere Middleware-Komponente, für Analyse oder Reporting wird eine zusätzliche Datenhaltung aufgebaut, und mit jeder Erweiterung entstehen neue Zuständigkeiten, Schnittstellen und Betriebsroutinen. So entsteht eine Landschaft, die lokal durchaus optimiert sein kann, sich als Gesamtsystem jedoch immer schwerer steuern lässt.

Nicht die Vielfalt einer Tool-Landschaft an sich macht sie zum strategischen Problem, sondern die Wechselwirkungen zwischen ihren Bestandteilen.

Typische Auslöser sind:

- **steigende Integrationsdichte**, weil immer mehr Systeme, Datenquellen und Prozesse zuverlässig zusammenspielen müssen
- **Performance-Grenzen**, die nicht in einem einzelnen System liegen, sondern an Übergängen, Datenbewegungen und zusätzlichen Verarbeitungsschichten entstehen
- **Governance-Probleme**, weil Zuständigkeiten, Sicherheitsmodelle, Compliance-Anforderungen und Betriebslogiken über mehrere Werkzeuge verteilt sind
- **Lizenz- und Herstellerabhängigkeiten**, die nicht nur Kosten beeinflussen, sondern auch Release-Zyklen, Roadmaps und Modernisierungspfade
- **personelle Abhängigkeiten**, weil spezialisiertes Wissen über mehrere Technologien, Layer und Einzellösungen verteilt wird

In der Anwendung von Künstlicher Intelligenz offenbart sich die Tragfähigkeit einer Systemlandschaft.

Diese Dynamik verschärft sich derzeit in vielen Unternehmen durch einen zusätzlichen Faktor: **Künstliche Intelligenz**. Gerade an dieser Stelle lässt sich die Tragfähigkeit einer Systemlandschaft erkennen. Denn was als KI-Initiative beginnt, wird schnell zur Frage, wie viele Datenkopien, Integrationspfade, Sicherheitsgrenzen und Medienbrüche überwunden werden müssen, bevor aus Daten überhaupt belastbare Arbeitsgrundlagen werden. Hinzu kommt der oft unterschätzte Aspekt der Nachvollziehbarkeit. Wer welche Daten wann in welchem Zustand verarbeitet hat, muss im laufenden Betrieb rekonstruierbar sein: Nicht nur für Compliance, sondern für jede belastbare Fehleranalyse. InterSystems IRIS speichert Nachrichten und Verarbeitungszustände persistent. Jede Transformation bleibt dokumentiert, jeder Zustand ist reproduzierbar. Das ist kein Logging-Feature, sondern ein architektonisches Prinzip.

In gewachsenen Landschaften steigt der Aufwand dann selten linear. Stattdessen wächst die Zahl der Übergänge, Sonderlogiken und Abstimmungsbedarfe überproportional. Sichtbar wird das vor allem an drei Stellen: im Betrieb, in der Weiterentwicklung und in der Steuerung der gesamten Systemlandschaft. Störungen lassen sich nicht mehr klar einem einzelnen System zuordnen, neue Anforderungen müssen über bestehende Tool-Grenzen hinweg eingefügt werden, und Verantwortlichkeiten verteilen sich zunehmend auf mehrere Teams und Werkzeuge. Die eigentliche Herausforderung liegt dann nicht mehr in der Leistungsfähigkeit einzelner Komponenten, sondern in der Frage, wie viel Energie in die Steuerung fließen muss, um das Gesamtsystem beherrschbar zu halten.

Ein weiterer Kipppunkt ist erreicht, wenn sich die Bestandteile des Technologie-Stacks nicht mehr gleichmäßig weiterentwickeln. Unterschiedliche Release-Zyklen, zeitversetzt adressierte Sicherheitslücken, abweichende Roadmaps und inkompatible Modernisierungspfade erhöhen nicht nur den Betriebsaufwand, sondern auch das Risiko, dass Innovation und Stabilität gegeneinander ausgespielt werden. Dann wird aus einem technisch gewachsenen Stack ein Managementproblem. Entscheidungen verzögern sich, weil jede Änderung zuerst auf Abhängigkeiten, Betriebsfolgen und Sicherheitsimplikationen geprüft werden muss.

Woran CIOs den Kippunkt erkennen

Ein Technologie-Stack wird strategisch kritisch, wenn mehrere der folgenden Symptome gleichzeitig auftreten:

- neue Anforderungen lösen fast automatisch neue Schnittstellen, Zusatztools oder Datenkopien aus
- die Ursachenanalyse verläuft über mehrere Teams, Systeme und Zeitfenster hinweg
- Sicherheits- und Release-Management werden durch unterschiedliche Zyklen einzelner Werkzeuge gebremst
- Betriebsstabilität hängt spürbar vom Erfahrungswissen einzelner Personen ab
- Innovationsvorhaben in Analytics, Automatisierung oder KI scheitern nicht an der Idee, sondern an den architektonischen Vorarbeiten
- Viele Systeme oder Datenquellen müssen dauerhaft zusammenspielen. Zudem ist absehbar, dass in den kommenden Jahren weitere hinzukommen.
- Sicherheits-, Release- und Governance-Aufwand steigen mit jedem zusätzlichen Werkzeug überproportional.
- Integrationsdichte, betriebliche Kritikalität und Anforderungen an verlässliche Daten nehmen weiter zu.
- Zusätzliche Managementfrage: Wie oft wird eine fachliche Entscheidung heute durch architektonische Reibung verlangsamt?

Je mehr dieser Punkte erfüllt sind, desto sorgfältiger sollte geprüft werden, ob die bestehende Stack-Logik noch trägt oder ob ein plattformbasierter Ansatz strategisch sinnvoller ist.

Die größten Kosten und größten Steuerungsprobleme entstehen in komplexen Umgebungen an den Übergängen zwischen verschiedenen Software-Tools.

1.2 Technologie-Stack oder Plattform: Welche Architektur ist tragfähiger?

Die Alternative zwischen Technologie-Stack und Plattform wird oft zu schematisch diskutiert. In der Praxis geht es nicht um ein abstraktes „modular versus integriert“, sondern um die Frage, welche Form von Komplexität ein Unternehmen beherrschen kann. Ein Stack optimiert Teilprobleme. Eine Plattform reduziert Übergänge zwischen ihnen. Welche Logik sinnvoller ist, hängt deshalb weniger von technologischen Präferenzen ab als von der Anzahl der eingesetzten Werkzeuge, dem Grad ihrer Verknüpfung, der Rolle verlässlicher Daten für Prozesse und Entscheidungen, dem Betriebsmodell und der Häufigkeit fachlicher, regulatorischer oder technischer Änderungen.

Ein spezialisierter Technologie-Stack hat klare Stärken. Er folgt häufig einem Best-of-Breed-Ansatz: Für einzelne Anforderungen werden jeweils die Werkzeuge ausgewählt, die in ihrem Bereich funktional besonders stark sind. Das schafft Freiheitsgrade bei Technologieentscheidungen und kann in klar abgegrenzten Szenarien wirtschaftlich sehr effizient sein. Das gilt insbesondere dann, wenn ein Anwendungsfall fachlich eng umrissen ist, nur wenige Schnittstellen benötigt, eine überschaubare betriebliche Kritikalität hat und voraussichtlich nicht stark ausgebaut wird. Für einen kleinen, klar begrenzten Use Case kann ein Plattformansatz deshalb tatsächlich überdimensioniert sein.

Sobald Datenhaltung, Verarbeitung, Integration, Monitoring und Betrieb eng miteinander gekoppelt sind, verschiebt sich der Bewertungsmaßstab. Dann reicht es nicht mehr, für jede Teilaufgabe ein gutes Werkzeug zu haben. Maßgeblich wird, wie hoch der Aufwand an den Übergängen ist. Genau dort liegen in komplexeren Umgebungen oft nicht nur die größten Kosten, sondern auch die größten Steuerungsprobleme.

Für die Abwägung zwischen Stack und Plattform sind vor allem diese Kriterien relevant:

- **Integrationsdichte:** Wie viele Systeme und Datenquellen müssen dauerhaft zusammenspielen?
- **Betriebsanforderungen:** Wie hoch sind Verfügbarkeits-, Transparenz- und Sicherheitsanforderungen?
- **Änderungsdruck:** Wie häufig verändern sich fachliche, regulatorische oder technische Anforderungen?
- **Strategische Reichweite der Architektur:** Geht es um einen engen Anwendungsfall oder um einen tragfähigen Kern für mehrere Prozesse und Datenflüsse?
- **Entscheidungsgeschwindigkeit:** Wie stark verzögern architektonische Übergänge die Umsetzung neuer Anforderungen?
- **Wissensmanagement:** Wie viel Fachpersonal ist notwendig, um die Infrastruktur zu betreiben und wie wird das Wissen zu Prozessen innerhalb des Teams weitergegeben?

In einem Stack genügt es nicht, dass jedes einzelne Tool fachlich funktioniert. Es muss sichergestellt sein, dass alle beteiligten Komponenten rechtzeitig adressiert werden.

Ab einer bestimmten Integrations- und Betriebsdichte reicht die Betrachtung einzelner Werkzeuge oft nicht mehr aus. Skalierung ist dabei kein einheitliches Konzept. InterSystems IRIS adressiert das mit einer Architektur, die beide Wege strukturell unterstützt: Einzelinstanzen können vertikal wachsen, also durch zusätzliche Ressourcen auf demselben System. Für horizontale Skalierung bietet InterSystems IRIS das Enterprise Cache Protocol (ECP): Es verteilt Anwendungslast auf mehrere Applikationsserver, während der Datenbankserver zentral den Zustand hält, ohne dass die Anwendungslogik dafür angepasst werden muss. Neue Server lassen sich bei wachsender Last hinzufügen. Fällt einer aus, können Nutzer nach dem Reconnect weiterarbeiten. Fällt der Datenbankserver aus, halten die Applikationsserver den Zustand vor und versuchen, die Verbindung wiederherzustellen. Hier ist entscheidend, welcher Skalierungspfad im eigenen Betriebsmodell realistisch ist und dass die Architektur so aufgebaut ist, dass sie diesen Pfad nicht blockiert, sondern aktiv ermöglicht.

1.3 Wirtschaftliche Bewertung und Risikoabwägung

Architekturentscheidungen werden noch immer zu häufig auf sichtbare Lizenz- oder Anschaffungskosten verengt. Das greift zu kurz. Für die tatsächliche Wirtschaftlichkeit einer Systemlandschaft sind oft andere Faktoren entscheidender: Anbindungsaufwand, Betriebsaufwand, Sicherheitsmanagement, Fehlersuche, Spezialwissen, Governance und der Aufwand, unterschiedliche Produkt- und Release-Zyklen überhaupt aufeinander abgestimmt zu halten.

Gerade in gewachsenen Stacks verteilen sich diese Kosten über viele Stellen. Jedes zusätzliche Werkzeug kann für sich genommen sinnvoll sein und dennoch den Gesamtaufwand erhöhen. Besonders relevant wird das, wenn Sicherheitsaspekte und Weiterentwicklungsfähigkeit mit betrachtet werden. In einem Stack genügt es nicht, dass jedes einzelne Tool fachlich funktioniert. Es muss auch sichergestellt werden, dass Sicherheitslücken in allen beteiligten Komponenten rechtzeitig adressiert werden, Patches und Updates nicht in neuen Inkompatibilitäten resultieren und sich die Werkzeuge in ähnlichem Tempo weiterentwickeln oder zumindest kompatibel bleiben.

Gerade an dieser Stelle wird deutlich, dass Werkzeuge innerhalb eines Stacks nicht gleichmäßig altern. Einige werden aktiv weiterentwickelt, andere stagnieren. Manche Hersteller schließen Schwachstellen schnell, andere mit Verzögerung. Manche Produkte passen ihre Roadmap an neue Anforderungen an, andere bleiben hinter den Erwartungen

des Gesamtprojekts zurück. Daraus entsteht nicht nur zusätzlicher Betriebsaufwand, sondern ein reales Architektur- und Sicherheitsrisiko. Denn mit jeder zusätzlichen Komponente steigt die Wahrscheinlichkeit, dass ein einzelnes Element zum begrenzenden Faktor der Gesamtarchitektur wird.

Bewertungsrahmen – sieben Kosten- und Risikoblöcke für die Architekturbewertung

Für eine belastbare Bewertung ist ein Gesamtmodell sinnvoller als der isolierte Vergleich einzelner Werkzeuge. Ein praxistauglicher Rahmen umfasst insbesondere sieben Kosten- und Risikoblöcke:



Die wirtschaftliche Leitfrage verschiebt sich damit weg vom Einzelpreis und hin zur Architekturbeherrschung: Ausschlaggebend ist die Frage, was es kostet, die gesamte Systemlandschaft unter realen Anforderungen stabil, sicher und anschlussfähig zu halten.

Daraus ergeben sich in der Praxis zwei unterschiedliche Bewertungsprofile. Ein Stack kann zunächst geringere oder auf mehrere Werkzeuge verteilte Einstiegskosten aufweisen, erzeugt jedoch häufig zusätzlichen Aufwand an Schnittstellen, Zuständigkeiten, Sicherheitsgrenzen und Release-Abhängigkeiten. Eine Plattform macht ihre Kernkosten meist sichtbarer, kann dafür aber Integrationsaufwand, Transparenzdefizite, Sicherheitskomplexität und betriebliche Reibung reduzieren.

Die wirtschaftliche Leitfrage verschiebt sich damit weg vom Einzelpreis und hin zur Architekturbeherrschung: Nicht das einzelne Werkzeug ist ausschlaggebend, sondern die Frage, was es kostet, die Systemlandschaft unter realen Anforderungen stabil, sicher und anschlussfähig zu halten. Wirtschaftlich sinnvoll ist eine Architektur nicht dann, wenn sie den niedrigsten sichtbaren Einstiegspreis hat, sondern dann, wenn sie den Gesamtaufwand im laufenden Betrieb und unter Veränderungsdruck dauerhaft beherrschbar hält. Für Unternehmen ist das der grundlegende Vergleichsmaßstab, weil sich genau dort Investitionssicherheit und Folgekosten eines technologischen Konzepts entscheiden.



2 Datenarchitektur als Kernentscheidung

Datenarchitektur bestimmt, wie hoch der Aufwand für Integration, Auswertung, Weiterentwicklung und Betrieb im Alltag tatsächlich ist. Wie Daten organisiert, verbunden und nutzbar gemacht werden, entscheidet über die Tragfähigkeit der gesamten Systemlandschaft.

2.1 Abhängigkeiten und strategische Steuerbarkeit

In der Praxis wird Lock-in oft zu eng verstanden, nämlich als Bindung an einen einzelnen Hersteller oder eine Plattform. Für die langfristige Architekturstrategie ist das jedoch nur ein Teil des Problems.

IT-Entscheidungen erzeugen immer Abhängigkeiten. Die Frage ist nicht, ob sie entstehen, sondern welche Form sie annehmen und wie stark sie die Handlungsfähigkeit der Organisation beeinflussen. In der Praxis wird Lock-in oft zu eng verstanden, nämlich als Bindung an einen einzelnen Hersteller oder eine Plattform. Für die langfristige Architekturstrategie ist das jedoch nur ein Teil des Problems. Genauso relevant sind Abhängigkeiten, die aus Datenkopien, Integrationsbrücken, Spezialwerkzeugen, verteilten Sicherheitsmodellen und lokal gewachsenen Betriebsroutinen entstehen. Lock-in entsteht damit nicht nur durch Herstellerbindung, sondern auch durch die strukturelle Bindung an eine fragmentierte Systemlandschaft.

Gerade in heterogenen Landschaften wächst diese Form der Bindung selten durch eine große Grundsatzentscheidung, sondern durch viele kleine Erweiterungen. Einzelne Ergänzungen sind oft plausibel. In der Summe verlagert sich die Weiterentwicklung jedoch auf die Übergänge zwischen Werkzeugen, Datenhaltungen und Zuständigkeiten.

Damit verändert sich auch die Frage, worin der strategische Wert einer Plattform liegt. Strategisch relevant ist eine Plattform nicht deshalb, weil sie möglichst viele Funktionen in sich vereint. Maßgeblich ist vielmehr, ob sie zusätzliche Kernabhängigkeiten vermeidet. Wenn für jede neue Anforderung eine weitere Basistechnologie eingeführt werden muss, nimmt die Steuerbarkeit fast zwangsläufig ab.

In historisch gewachsenen Systemlandschaften entsteht eine Distanz zwischen Entstehung und Nutzung von Daten – mit Folgen für Konsistenz, Risiko und Wartbarkeit.

Wenn neue Anforderungen dagegen innerhalb eines konsistenten Kerns aufgenommen werden können, kann die architektonische Beweglichkeit steigen – auch dann, wenn damit eine bewusst in Kauf genommene Bindung an eine Plattform verbunden ist. Aus Entscheidungssicht ist genau das der richtungsweisende Unterschied: problematisch ist nicht jede Bindung, sondern ein architektonischer Rahmen, der mit jeder Erweiterung neue technische und organisatorische Bindungen erzeugt.

Für die strategische Steuerbarkeit sind deshalb vor allem drei Fragen relevant:

- **Entstehen neue Anforderungen** innerhalb eines bestehenden Architektur- und Datenkerns oder regelmäßig über zusätzliche Basistechnologien?
- **Liegen die Abhängigkeiten** an klaren Plattformgrenzen oder verteilen sie sich über viele Integrations-, Sicherheits- und Betriebsübergänge?
- **Bleibt die Gesamtarchitektur** auch dann noch steuerbar, wenn Integrationsdichte, Datenvolumen und regulatorische Anforderungen weiter zunehmen?

Wo diese Fragen nicht mehr belastbar beantwortet werden können, sinkt die strategische Steuerbarkeit. Dann ist nicht nur die Bindung an einen Hersteller relevant, sondern die Bindung an die eigene Fragmentierung. Die eigentliche Frage lautet dann nicht mehr „Wie offen ist unsere Architektur?“, sondern „Wie viele neue Grundsatzentscheidungen müssen wir für jede Erweiterung erneut treffen?“

Was Lock-in aus Managementsicht wirklich bedeutet

Lock-in sollte in diesem Kontext bewusst breiter verstanden werden. Strategisch kritisch wird nicht nur die Bindung an einen Hersteller, sondern auch die Bindung an eine Architektur, die auf Dauer nur über zusätzliche Kopien, Integrationsroutinen, Spezialwissen und Sonderlogiken weiterentwickelt werden kann.

Für Führungskräfte sind deshalb drei Formen von Abhängigkeit zu unterscheiden:



Hersteller-
abhängigkeit



Abhängigkeit von
Spezialwissen und
Einzellösungen



Abhängigkeit von der eigenen
historisch gewachsenen
Fragmentierung

Die dritte Form ist im laufenden Betrieb oft die teuerste, weil sie Innovationsfähigkeit und Steuerbarkeit schleichend begrenzt, ohne zunächst als klassische Herstellerbindung sichtbar zu werden.

2.2 Datenbewegung versus Datennähe

Viele Architekturprobleme erscheinen zunächst als Integrations-, Performance- oder Betriebsprobleme, obwohl ihre Ursache tiefer liegt: in der Art, wie Daten durch die Systemlandschaft bewegt werden. Gemeint ist damit mehr als der reine Transfer zwischen Systemen. Datenbewegung umfasst auch Extraktion, Transformation, Zwischenspeicherung, Neuablage und die erneute Bereitstellung in einem anderen fachlichen oder technischen Kontext. Sobald dieses Muster zur Standardantwort auf neue Anforderungen wird, verschiebt sich die Architektur weg von einem belastbaren Datenkern hin zu einer Kette von Übergängen.

Statt Daten für jeden neuen Zweck erneut zu verschieben, zu transformieren und an anderer Stelle vorzuhalten, bleibt ein gemeinsamer Kern erhalten, auf den unterschiedliche Nutzungskontexte zugreifen.

In historisch gewachsenen Systemlandschaften ist genau das häufig zu beobachten. Operative Systeme erzeugen Daten, Integrationsschichten bereiten sie für andere Prozesse auf, Analyseumgebungen führen eigene Bestände, Reporting- und Automatisierungslösungen arbeiten mit weiteren Ableitungen. Jeder dieser Schritte kann funktional sinnvoll sein. Zusammen entstehen jedoch zusätzliche Synchronisationspunkte und eine wachsende Distanz zwischen Entstehung und Nutzung von Daten – mit Folgen für Konsistenz, Risiko und Wartbarkeit.

Die Konsequenzen sind strukturell:

- **Konsistenzprobleme**, weil mehrere Datenstände, Transformationspfade und Aktualitätsniveaus parallel gepflegt werden müssen
- **höheres Risiko**, weil jeder zusätzliche Übergang neue technische, fachliche und sicherheitsrelevante Fehlerstellen schafft
- **schlechtere Wartbarkeit**, weil Änderungen an Datenmodellen oder Prozessen an mehreren Stellen nachvollzogen und nachgezogen werden müssen

Datennähe setzt an genau diesem Punkt an. Das Prinzip ist einfach: Daten sollen möglichst nah an den Kontexten bleiben, in denen sie verarbeitet, integriert und ausgewertet werden. Dadurch sinkt nicht nur der Integrations- und Verwaltungsaufwand. Auch Aktualität, Reaktionsgeschwindigkeit und Wirtschaftlichkeit können sich verbessern, weil Daten nicht für jeden zusätzlichen Zweck über neue Strecken verschoben, mehrfach gespeichert oder in separaten, häufig cloudbasierten, Umgebungen vorgehalten werden müssen.

Das bedeutet nicht, dass in heterogenen Umgebungen keine Daten mehr bewegt werden. Es bedeutet aber, dass Datenbewegung nicht das dominierende Architekturprinzip bleibt.

Eine gemeinsame Datenbasis bedeutet, dass operative Prozesse, Integration und Auswertung nicht jeweils auf eigenen Datenkopien und separaten Grundstrukturen aufsetzen müssen. Statt Daten für jeden neuen Zweck erneut zu verschieben, zu transformieren und an anderer Stelle vorzuhalten, bleibt ein gemeinsamer Kern erhalten, auf den unterschiedliche Nutzungskontexte zugreifen. Das reduziert Abstimmungsaufwand, senkt Konsistenzrisiken und erleichtert es, neue Anforderungen auf bestehende Datenzusammenhänge aufzusetzen. Managementseitig lässt sich das einfach übersetzen; Daten stehen dort zur Verfügung, wo sie gebraucht werden, ohne dass für jeden zusätzlichen Zweck erst eine neue Datenstrecke organisiert werden muss.

Damit stehen Daten nicht erst verzögert oder nur in abgeleiteten Beständen zur Verfügung, sondern in einem Zustand, der fachlich belastbar, technisch anschlussfähig und zeitlich aktuell genug ist, um neue Anforderungen ohne lange Ketten von Vorverarbeitung und Umkopieren zu unterstützen. Genau hier entscheidet sich, ob eine Architektur Erweiterungen aufnehmen kann oder ob jede neue Anforderung zusätzliche Übergänge, Kopien und Abstimmungsaufwand erzeugt.

Sobald neue Anforderungen fast immer neue Kopier- und Übergabepfade erzeugen, steigt der Aufwand. Wo Datennähe dagegen zum Architekturprinzip wird, sinkt die Zahl der Brüche und damit auch die Last, die spätere Erweiterungen mit sich bringen.

Für die Architekturökonomie ist das ein zentraler Punkt. Je stärker Datenbewegung zum Normalzustand wird, desto höher ist der Aufwand, Daten fachlich, technisch und betrieblich konsistent zu halten. Wo dagegen ein gemeinsamer Datenkern unterschiedliche Nutzungskontexte tragen kann, werden Erweiterungen wahrscheinlicher als Weiterentwicklung eines bestehenden Zusammenhangs umgesetzt und nicht als zusätzliche Daten- und Integrationsstrecke.

Die eigentliche Frage lautet deshalb nicht, ob Datenbewegung vollständig vermieden werden kann. Ausschlaggebend ist, ob sie in der Architektur Ausnahme oder Normalzustand ist. Sobald neue Anforderungen fast immer neue Kopier- und Übergabepfade erzeugen, steigt der Aufwand zwangsläufig. Wo Datennähe dagegen zum Architekturprinzip wird, sinkt die Zahl der Brüche und damit auch die Last, die spätere Erweiterungen mit sich bringen. Damit wird eine datenzentrierte Architektur (Data-Centric Compute) greifbar: nicht als technisches Schlagwort, sondern als Architekturprinzip, das Datenkontext, Verarbeitung und Nutzung enger zusammenführt.

Was eine gemeinsame Datenbasis konkret bedeutet

Eine gemeinsame Datenbasis heißt nicht, dass alle Systeme verschwinden oder jede Datenbewegung überflüssig wird. Sie bedeutet, dass neue Anforderungen nicht automatisch neue Kopien, neue Speicherorte und neue Abstimmungswege erzwingen.

Das hat vier direkte Folgen:

- weniger Synchronisationsaufwand
- weniger Konsistenzrisiken
- kürzere Wege von operativer Nutzung zu Analyse und Reporting
- höhere Anschlussfähigkeit für neue Anforderungen, etwa in Analytics, Automatisierung oder KI

Die wirtschaftliche Relevanz liegt damit nicht nur in besserer Performance, sondern vor allem in geringerer struktureller Reibung.

2.3 Multi-Model: Relevanz und Grenzen

Ein Multi-Model-Ansatz erlaubt es, unterschiedliche Daten- und Zugriffsmuster auf einem gemeinsamen technologischen Kern abzubilden; etwa transaktionale Verarbeitung, Integrationslogik, dokumentenorientierten Zugriff oder analytische Nutzung. Relevant wird das dort, wo dieselben Daten in mehreren fachlichen und technischen Kontexten gebraucht werden und zusätzliche Spezialtechnologien vor allem neue Kopien, Synchronisationslogiken und Betriebsaufwand erzeugen würden.

Mit Basistechnologien sind hier grundlegende technische Bausteine gemeint, auf denen eigene Daten- oder Verarbeitungslogiken aufsetzen, etwa separate Datenbanken, Analyse-Engines, Integrationslaufzeiten oder spezialisierte Speicher. Kann eine Plattform mehrere dieser Funktionen in einem gemeinsamen Kern bündeln, lassen sich zusätzliche Übergänge zwischen solchen Bausteinen vermeiden.

Für Architektur- und IT-Verantwortliche ist das vor allem aus zwei Perspektiven relevant: Produktivität und Integration. Wenn operative Prozesse, Integrationsflüsse und analytische Nutzung jeweils auf eigenen Basistechnologien aufbauen, steigt der Abstimmungs-

Wenn dieselbe Plattform operative Verarbeitung, Integrationslogik und Analyse auf einem gemeinsamen Datenkontext abbilden kann, sinkt der Aufwand für Datenabgleich, Erweiterung und technische Koordination.

und Pflegeaufwand schnell. Wenn dagegen dieselbe Plattform mehrere Zugriffsmuster auf denselben Datenkontext zulässt, lassen sich neue Anforderungen häufiger als Erweiterung eines vorhandenen Kerns umsetzen statt als zusätzlicher technologischer Sonderfall. Genau darin liegt der strategische Nutzen eines Multi-Model-Ansatzes. Er reduziert nicht nur Technologievielfalt, sondern auch die Zahl der Übergänge, an denen Projekte, Verantwortlichkeiten und Betriebsmodelle auseinanderlaufen.

Ein typisches Beispiel ist eine Architektur, in der Auftragsdaten in einem operativen System verarbeitet, parallel über Integrationslogik an nachgelagerte Anwendungen weitergegeben und gleichzeitig für Reporting oder Forecasting ausgewertet werden. Wenn dieselbe Plattform operative Verarbeitung, Integrationslogik und Analyse auf einem gemeinsamen Datenkontext abbilden kann, sinkt der Aufwand für Datenabgleich, Erweiterung und technische Koordination deutlich.

Ob ein Multi-Model-Ansatz sinnvoll ist, lässt sich bereits vor der Einführung prüfen. Entscheidungsrelevant ist, ob dieselben Daten voraussichtlich in mehreren Nutzungskontexten gebraucht werden, ob neue Anforderungen sonst zusätzliche Spezialtechnologien erfordern würden und ob der Mehraufwand vor allem an Übergängen, Kopien und Betriebslogiken entstünde. Wo das der Fall ist, spricht viel dafür, zusätzliche Anforderungen nicht über neue technische Bausteine, sondern über einen gemeinsamen Kern abzubilden.

Wann Multi-Model wirklich relevant wird

Ein Multi-Model-Ansatz ist vor allem dann sinnvoll, wenn dieselben Daten gleichzeitig in mehreren Rollen gebraucht werden, zum Beispiel:

- für operative Verarbeitung
- für Integration und Prozesslogik
- für Reporting, Forecasting oder Analyse

Wird jede dieser Anforderungen sonst durch eine neue Basistechnologie beantwortet, steigt die Komplexität nicht nur technisch, sondern auch organisatorisch.

Nicht sinnvoll ist Multi-Model als Selbstzweck. Wenn nur ein einzelner, klar begrenzter Anwendungsfall mit geringer Integrationsdichte betrachtet wird, erscheint eine breitere Plattformsentscheidung fast immer unnötig.

Multi-Model ist vor allem in Systemlandschaften mit folgenden Merkmalen relevant:

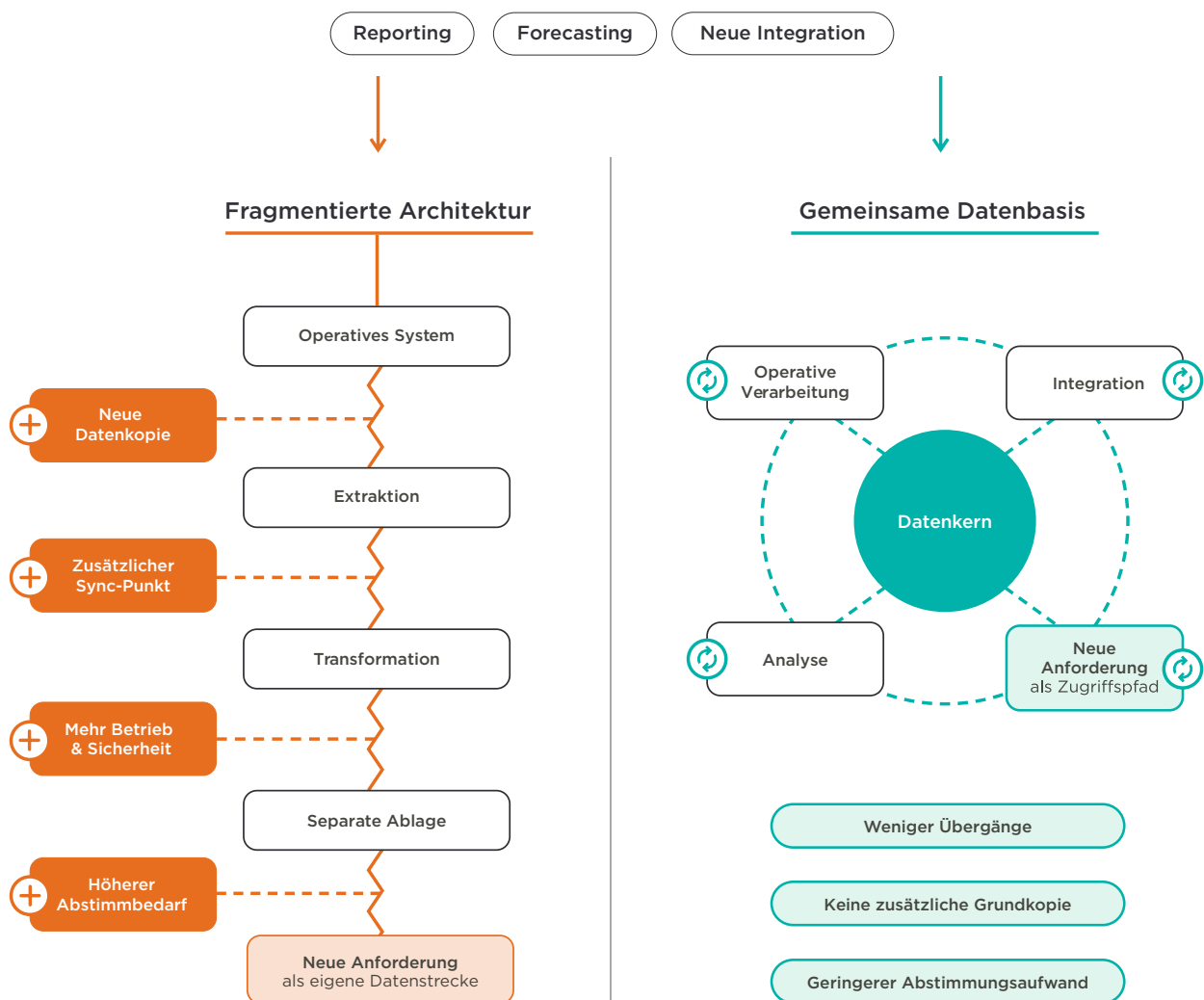
- **dieselben Daten** werden in mehreren Nutzungskontexten gebraucht
- **operative Verarbeitung**, Integration und Auswertung greifen enger ineinander
- **zusätzliche Basistechnologien** würden vor allem neue Übergänge, Kopien und Betriebsmodelle erzeugen

Weniger zwingend ist der Ansatz dort, wo ein klar umrissener Anwendungsfall nur einen einzigen Zugriffspfad benötigt und keine nennenswerte architektonische Breite entstehen soll. Dann wäre es überzogen, aus einer begrenzten Fachanforderung unmittelbar eine Grundsatzentscheidung über den Plattformkern abzuleiten.

Multi-Model ist damit keine Feature-Sammlung und keine Buzzword-Debatte, sondern eine Architekturentscheidung gegen unnötige Grundsatzvielfalt. Die eigentliche Frage lautet nicht, wie viele Modelle eine Plattform unterstützt, sondern wie viele zusätzliche Basistechnologien eingeführt werden müssen, um Daten in verschiedenen Kontexten sinnvoll nutzbar zu machen.

Was eine neue Anforderung architektonisch auslöst

Neue fachliche Anforderung:





3 Entwicklung, Umsetzung und risikoarmer Einstieg

Architekturentscheidungen bestimmen nicht nur den späteren Betrieb, sondern bereits die Umsetzung selbst: Projektlaufzeiten, Abstimmungsaufwand und die Geschwindigkeit, mit der aus Anforderungen belastbare Ergebnisse entstehen. Besonders in Entscheidungspositionen ist es relevant, dass sich der Nutzen einer Architektur nicht erst im Zielbild, sondern schon im Weg dorthin zeigt.

Architektur beeinflusst unmittelbar die Zeit bis zum Einsatz im operativen Umfeld, also die Time-to-Value. Je länger dieser Zeitraum ist, desto höher sind die Opportunitätskosten.

In vielen gewachsenen Systemlandschaften müssen Anforderungen über mehrere Schichten hinweg umgesetzt werden. Schnittstellen werden abgestimmt, Daten zwischen Systemen bewegt und Verantwortlichkeiten verteilt. Jede Änderung betrifft damit nicht nur eine fachliche Logik, sondern eine Kette von Abhängigkeiten. Für die Umsetzung bedeutet das: Nicht einzelne Entwicklungsschritte entscheiden über das Tempo, sondern die Zahl der Übergänge, die fachlich, technisch und organisatorisch bewältigt werden müssen. Je weniger Übergänge es zu koordinieren gibt, desto direkter wird der Weg von der Anforderung zur Umsetzung.

Maßgeblich ist dabei nicht nur, wie schnell ein Projekt abgeschlossen wird, sondern vor allem, wie schnell daraus ein messbarer Effekt entsteht. In vielen Organisationen liegt die eigentliche Verzögerung nicht in der Entwicklung selbst, sondern in nachgelagerten Koordinationsschleifen, Integrationsaufwänden und Anpassungen im Betrieb. Architektur beeinflusst damit unmittelbar die Zeit bis zum wirksamen Einsatz im operativen Umfeld, also die Time-to-Value eines Vorhabens. Je länger dieser Zeitraum ist, desto höher sind auch die Opportunitätskosten. Prozesse bleiben länger ineffizient, Automatisierung verzögert sich und neue Anforderungen können nicht im gleichen Tempo umgesetzt werden, in dem sie entstehen. Architektur wird damit zu einem wirtschaftlichen Faktor, nicht nur zu einer technischen Entscheidung.

InterSystems IRIS adressiert genau diese strukturelle Komplexität, indem Datenmanagement, Integration und Ausführungslogik in einer gemeinsamen Plattform zusammengeführt werden. Im Unterschied zu anderen Lösungen, in denen Datenmanagement, Integration und Ausführungslogik über mehrere technische Schichten hinweg organisiert werden,

führt die Plattform diese Funktionen enger zusammen. Anforderungen können dadurch häufiger in einem konsistenten technischen Kontext umgesetzt werden, ohne zusätzliche Übergänge zwischen separaten Systemen organisieren zu müssen. Das verändert nicht die fachliche Komplexität eines Projekts. Es verändert aber den Weg von der Anforderung bis zur produktiven Umsetzung. Der Koordinationsaufwand sinkt, Abhängigkeiten sind überschaubarer und Umsetzungsschritte klarer sequenzierbar. Gerade bei Integrationsszenarien, die sich kontinuierlich erweitern, wirkt sich dieser Unterschied direkt auf Projektlaufzeit und Time-to-Value aus.

3.1 Umsetzungsgeschwindigkeit als Ergebnis von Architektur

In fragmentierten Systemlandschaften entstehen häufig verdeckte Zusatzaufwände: Abstimmungen zwischen Teams, Anpassungen an Schnittstellen, Synchronisation von Datenbeständen oder zusätzliche Testschleifen.

Diese Faktoren sind selten isoliert sichtbar, summieren sich jedoch entlang des gesamten Projektverlaufs. Schon kleinere Änderungen können dadurch unverhältnismäßig viel Zeit in Anspruch nehmen, weil sie mehrere Systeme gleichzeitig betreffen. Gerade diese indirekten Effekte machen Architektur zu einer Managementfrage. Mit InterSystems IRIS entfällt ein großer Teil dieses Steuerungsaufwands. Anforderungen lassen sich dadurch direkter umsetzen, ohne dass jede Erweiterung neue technische Verbindungen nach sich zieht.

Mit InterSystems IRIS entfällt ein großer Teil dieses Steuerungsaufwands. Anforderungen lassen sich dadurch direkter umsetzen, ohne dass jede Erweiterung neue technische Verbindungen nach sich zieht.

Im konkreten Einsatz macht sich das vor allem in drei Bereichen bemerkbar:

- **Neue Schnittstellen und Prozesse** lassen sich deutlich schneller implementieren, weil weniger Systemgrenzen überwunden werden müssen.
- **Änderungen an bestehenden Abläufen** bleiben häufiger auf den betroffenen Funktionsbereich begrenzt, anstatt Anpassungen in mehreren abhängigen Systemen auszulösen.
- **Fehler lassen sich schneller identifizieren** und beheben, weil weniger Systemgrenzen analysiert werden müssen.

Der Produktivitätsgewinn entsteht dabei weniger durch punktuelle Beschleunigung einzelner Schritte als durch eine Architektur, die zusätzliche Schritte, Übergänge und Koordinationsaufwand vermeidet.

Darüber hinaus wirkt sich die Architektur auch auf die Fähigkeit aus, mehrere Vorhaben parallel umzusetzen. Wenn Integrationen, Datenflüsse und Logik eng miteinander verzahnt sind, entstehen weniger gegenseitige Abhängigkeiten zwischen Teams und Projekten. Neue Anforderungen blockieren sich seltener gegenseitig, weil sie nicht jeweils eigene technische Teilstrecken aufbauen müssen. Damit verschiebt sich die Frage von „Wie schnell ist ein einzelnes Projekt?“ zu „Wie viele relevante Vorhaben können gleichzeitig umgesetzt werden?“. Gerade in dynamischen Umfeldern wird diese Fähigkeit zu einem ausschlaggebenden Faktor für die Leistungsfähigkeit der IT.

Gleichzeitig bleibt der initiale Einarbeitungsaufwand bestehen. Neue Werkzeuge und Konzepte müssen verstanden und im Team verankert werden. Dieser Aufwand ist Teil jeder Technologieentscheidung. Bedeutsam ist deshalb nicht, ob dieser Aufwand existiert, sondern ob er sich im weiteren Projektverlauf amortisiert, etwa durch geringere Koordinationskosten, schnellere Erweiterbarkeit und stabilere Umsetzungen. Das breite Funktionsspektrum der Datenplattform InterSystems IRIS kann auf den ersten Blick komplex wirken. In der praktischen Umsetzung wird jedoch deutlich, dass gerade die Bündelung zentraler Funktionen den Einstieg nicht erschwert, sondern viele der sonst üblichen technischen Umwege vermeidet.

3.2

Einstieg und Validierung unter realen Bedingungen

Die Wirkung einer Architektur lässt sich nur begrenzt theoretisch bewerten. Entscheidend ist, ob sie im eigenen Kontext trägt – mit den vorhandenen Systemen, Datenstrukturen und organisatorischen Rahmenbedingungen.

Der Einstieg sollte bewusst nicht über eine umfassende Migration erfolgen, sondern über einen klar abgegrenzten Anwendungsfall. Ziel ist es, einen Anwendungsfall zu wählen, der einerseits beherrschbar ist, andererseits aber die typischen Integrations- und Datenanforderungen des Unternehmens abbildet. Auf dieser Basis lässt sich prüfen, ob sich Anforderungen mit InterSystems IRIS konsistent umsetzen lassen und wie sich der tatsächliche Aufwand im Vergleich zur bestehenden Architektur darstellt. Gleichzeitig wird sichtbar, wie sich die Plattform in bestehende Betriebsmodelle einfügt und welche Anpassungen organisatorisch notwendig sind.

Wichtig für eine tragfähige Entscheidung ist nicht, dass eine Plattform im Grundsatz überzeugt – sondern dass sie im konkreten Einsatzkontext nachweisbar funktioniert.

Deshalb ist ein Proof of Concept hier mehr als ein Demo-Format.

Ein solcher Einstieg ersetzt Annahmen durch belastbare Erkenntnisse. Insbesondere Integrationsaufwand, Stabilität im Betrieb und Erweiterbarkeit lassen sich vorab nur begrenzt bewerten. Erst unter realen Bedingungen wird sichtbar, wie sich die Architektur tatsächlich verhält. Damit reduziert sich das Entscheidungsrisiko erheblich. Fehlannahmen über Komplexität, Betriebsaufwand oder Skalierbarkeit werden früh erkennbar, bevor sie sich in größeren Projekten verfestigen. Gleichzeitig entsteht eine fundierte Grundlage für die Frage, ob und in welchem Umfang eine Plattform weiter ausgerollt werden sollte.

Ebenso wichtig ist die umgekehrte Perspektive. Wenn kein klarer Vorteil erkennbar ist oder der Aufwand den Nutzen übersteigt, wird auch das frühzeitig sichtbar. Wichtig für eine tragfähige Entscheidung ist nicht, dass eine Plattform im Grundsatz überzeugt – sondern dass sie im konkreten Einsatzkontext nachweisbar funktioniert. Deshalb ist ein Proof of Concept hier mehr als ein Demo-Format. Er ist ein begrenzter Realitätscheck für Architekturannahmen.

Vom begrenzten Einstieg zur belastbaren Entscheidung

Ein sinnvoller Einstieg folgt idealerweise einer klaren Logik:

- 1. einen fachlich relevanten, aber beherrschbaren Anwendungsfall wählen**
- 2. reale Daten- und Integrationspfade einbeziehen**
- 3. Lernkurve, Betriebsaufwand und Umsetzbarkeit sichtbar machen**
- 4. Ergebnisse fachlich, technisch und organisatorisch bewerten**
- 5. erst danach über Ausbau, Roll-out oder eine Überführung in eine Plattformarchitektur entscheiden**

So wird aus einem ersten Projekt kein isolierter Test, sondern eine belastbare Entscheidungsgrundlage.

Integration als Wachstumsmotor

Die Integrationskosten konnten um 57 Prozent gesenkt werden, während sich die Investition in kürzester Zeit amortisierte.



Ein international tätiger Logistikdienstleister mit stark wachsendem E-Commerce-Geschäft stand vor der Herausforderung, eine zunehmend komplexe Integrationslandschaft zu stabilisieren und gleichzeitig weiterzuentwickeln. Die bestehende Plattform verband mehrere hundert Schnittstellen zu Kunden, Partnern, Zollsystemen und internen Anwendungen. Mit wachsendem Geschäftsvolumen entwickelte sie sich zunehmend zum Engpass.

Änderungen an bestehenden Prozessen dauerten häufig mehrere Tage, neue Integrationen mehrere Wochen oder Monate. Gleichzeitig erschwerte fehlende End-to-End-Transparenz die Fehleranalyse und den stabilen Betrieb. Grund für die Herausforderungen war der gewählte Low Code / No Code-Ansatz, der aufgrund komplexer Integrationsszenarien nur mit hohem Aufwand umgesetzt werden konnte.

Statt die bestehende Lösung weiter zu optimieren, entschied sich das Unternehmen für eine grundlegende Neuausrichtung seiner Integrationsarchitektur auf Basis von InterSystems IRIS. Ziel war es, Integrationsprozesse enger mit Datenverarbeitung und Anwendungslogik zu verzahnen und die Abhängigkeit von zusätzlichen Systemen zu reduzieren.

Die Umsetzung erfolgte schrittweise. Zunächst wurden besonders komplexe und geschäftskritische Integrationen migriert, darunter E-Commerce- und Tracking-Prozesse. Weitere Schnittstellen werden seitdem kontinuierlich übertragen. Bereits im laufenden Betrieb zeigten sich deutliche Effekte. Prozesse, die zuvor mehrere Minuten oder Stunden benötigten, werden heute in Sekunden ausgeführt. Die Erstellung von Tracking-Reports verkürzte sich von etwa 15 Minuten auf unter 30 Sekunden.

Auch die Entwicklung neuer Integrationen beschleunigte sich deutlich. Statt Monaten lassen sich neue Schnittstellen heute meist innerhalb weniger Tage bis Wochen umsetzen. Gleichzeitig reduzierte sich der Abstimmungsaufwand, da Integrationen mit kleineren, klar definierten Teams umgesetzt werden können. Neben den operativen Verbesserungen ergaben sich auch wirtschaftliche Effekte. Die Integrationskosten konnten um 57 Prozent gesenkt werden, während sich die Investition in kürzester Zeit amortisierte.

Die neue Architektur schafft damit einen tragfähigeren Kern für weitere Automatisierungs-, Analyse- und perspektivisch auch KI-nahe Anwendungsfälle.



4 Performance, Skalierung und Stabilität

Nach der Frage, wie sich Architektur auf Umsetzung und Einstieg auswirkt, rückt nun das Verhalten im laufenden Betrieb in den Vordergrund. Performance ist dabei nicht allein mit Geschwindigkeit gleichzusetzen. Für die Architekturentscheidung ist wichtiger, wie sich eine Plattform unter wachsender Last verhält, wie sie mit steigenden Datenmengen umgeht und ob sie auch dann stabil und performant bleibt, wenn Deployments über mehrere Systeme, Umgebungen oder Standorte hinweg organisiert werden.

Eine Plattform muss klein und wirtschaftlich beginnen können, darf spätere Wachstumsstufen aber nicht blockieren. Erst dann entsteht ein technischer Pfad, der auch unter veränderten Rahmenbedingungen Bestand hat.

Leistungsfähigkeit lässt sich nicht auf einzelne Benchmarkwerte reduzieren. Ein guter Einzelwert sagt wenig darüber aus, wie sich eine Plattform unter Parallelität, Lastspitzen oder wachsender Integrationsdichte verhält. Die Konsequenz: Wer eine Plattform nur im Idealfall testet, trifft keine belastbare Architekturentscheidung.

Performance, Skalierbarkeit und Stabilität sind dabei untrennbar miteinander verbunden. Eine Plattform ist nicht deshalb belastbar, weil sie in einem isolierten Szenario schnell reagiert, sondern weil sie auch bei steigenden Nutzerzahlen, wachsenden Datenmengen und höheren betrieblichen Anforderungen konsistent leistungsfähig bleibt. Genau hier setzt InterSystems IRIS an, indem diese Eigenschaften nicht getrennt, sondern innerhalb einer einheitlichen Architektur adressiert werden.

Darin liegt ein wesentlicher Unterschied für die Architekturentscheidung. Wachstum verändert die Anforderungen an Datenorganisation, Lastverteilung, Betriebsmodell und Ausfallsicherheit. Eine Plattform muss daher klein und wirtschaftlich beginnen können, darf spätere Wachstumsstufen aber nicht blockieren. Erst dann entsteht ein technischer Pfad, der nicht nur kurzfristig tragfähig ist, sondern auch unter veränderten Rahmenbedingungen Bestand hat. Die eigentliche Leitfrage lautet deshalb nicht, ob eine Plattform heute schnell genug ist, sondern ob sie auch morgen noch tragfähig skaliert und stabil betrieben werden kann.

4.1

Leistungsdimensionen technisch eingeordnet

Leistungsfähigkeit ist immer kontextabhängig und wird maßgeblich durch die Art der Last bestimmt. Nicht jedes Wachstumsproblem verlangt dieselbe Antwort. Wenn vor allem gleichzeitige Nutzer, Sessions oder Anwendungsprozesse bewältigt werden müssen, stellt sich eine andere Skalierungsfrage als bei schnell wachsenden Datenvolumina oder analytischen Abfragen über große Bestände. Performance lässt sich daher nur sinnvoll bewerten, wenn die zugrunde liegende Lastsituation berücksichtigt wird.

In der Praxis lassen sich drei Leistungsdimensionen unterscheiden:

- **Nutzer- und Prozesslast:** Steigt vor allem die Zahl gleichzeitiger Zugriffe, Sessions oder Anwendungsprozesse, die verarbeitet werden müssen, steht die Verteilung laufender Last im Vordergrund.
- **Datenvolumen und Datenzugriff:** Wachsen Datenmengen, Ingestion-Raten und analytische Abfragen über große Bestände, wird entscheidend, wie Daten über mehrere Systeme verteilt und effizient verarbeitet werden.
- **Betriebsmodell und Deployment:** Erfolgt Skalierung in verteilten, hybriden oder containerisierten Umgebungen, rückt die Frage in den Vordergrund, wie sich diese Last auch operativ stabil organisieren lässt.

Performance lässt sich nur sinnvoll bewerten, wenn die zugrunde liegende Lastsituation berücksichtigt wird.

Diese Dimensionen treten selten isoliert auf, bestimmen jedoch gemeinsam die Anforderungen an die Architektur. Für Unternehmen ist genau diese Unterscheidung wichtig, weil sich daraus ableitet, welcher Skalierungspfad überhaupt sinnvoll ist – mehr Hardware, verteilte Lastaufnahme oder eine grundlegend anders organisierte Datenhaltung.

Der naheliegende erste Schritt ist in vielen Umgebungen die vertikale Skalierung. Solange zusätzliche CPU- und Speicherkapazität wirtschaftlich sinnvoll genutzt werden kann, bleibt der Ausbau von Hardware-Ressourcen der einfachste Weg. Mit steigender Last stößt dieser Ansatz jedoch an seine strukturellen Grenzen. Wenn Last weiter steigt, ist entscheidend, ob eine Plattform darüber hinaus belastbar wächst und unterschiedliche Skalierungspfade unterstützt, ohne dass dafür ein zweiter Stack oder eine neue Datenbasis eingeführt werden muss.



**InterSystems
IRIS unterstützt
unterschiedliche
Skalierungspfade,
abhängig von der
Art der Last.**

InterSystems IRIS unterstützt unterschiedliche Skalierungspfade, abhängig von der Art der Last. Nimmt vor allem die Nutzer- und Prozesslast zu, kann diese über verteilte Caching-Modelle und die Trennung von Daten- und Applikationsservern horizontal abgefangen werden. Bleibt der Datenbestand dagegen selbst der begrenzende Faktor, verändert sich der Ansatz grundlegend. Dann geht es nicht mehr nur darum, Zugriffe auf vorhandene Daten effizient zu verteilen, sondern die Datenhaltung selbst horizontal zu organisieren. Für solche Szenarien wird horizontale Datenverteilung relevant, bei der Datenbestände über mehrere Server partitioniert werden.

Ein weiterer Punkt ist das Betriebsmodell. Skalierung findet heute nicht nur auf einem einzelnen Server statt, sondern zunehmend in verteilten, hybriden oder containerisierten Umgebungen. InterSystems IRIS lässt sich auch in solchen Szenarien betreiben und kann mit oder ohne Kubernetes sowohl als einzelne Instanz als auch in verteilten Strukturen ausgerollt werden. Damit bleibt Skalierung nicht auf ein einziges Deployment-Muster festgelegt, sondern lässt sich in unterschiedliche Betriebsrealitäten übersetzen.

Für die Architekturentscheidung ist das vor allem deshalb bedeutsam, weil sich Lastverhalten selten exakt prognostizieren lässt. Eine Plattform muss daher im größten denkbaren Zielbild überzeugen und gleichzeitig einen belastbaren Wachstumspfad eröffnen. Der Vorteil liegt dann nicht darin, von Anfang an maximal groß zu bauen, sondern klein zu beginnen und bei Bedarf kontrolliert zu erweitern. Genau darin liegt der strategische Wert. Wachstum wird nicht zum Anlass für eine neue Architekturgrundsatzentscheidung, sondern zum Ausbau eines bestehenden Kerns.

Was Skalierung aus Entscheidersicht bedeutet

Skalierung ist kein Selbstzweck und kein reines Technikthema. Für CIOs und IT-Entscheider:innen ist vor allem relevant, welche Form von Wachstum die Architektur aufnehmen muss:



**mehr gleichzeitige Nutzer
und Prozesse**



**größere Datenmengen und
höhere Ingestion-Raten**



**mehr analytische Last auf
bestehenden Beständen**



**komplexere Betriebsmodelle über
mehrere Umgebungen hinweg**

Erst aus dieser Einordnung ergibt sich, ob vertikale Skalierung ausreicht, verteilte Lastmodelle nötig werden oder die Datenhaltung selbst anders organisiert werden muss.

Skalierung unter realer Last



InterSystems IRIS setzt auf Mirroring als zentrales Hochverfügbarkeitsmodell. Der Grundgedanke dabei ist, dass ein gespiegelter Datenbestand auf einem zweiten System bereitsteht und im Fehlerfall automatisch übernommen werden kann.

Ein Beispiel dafür liefert eine internationale Investmentbank, die ihre bestehende Data-Lake-Umgebung für Echtzeitdaten, komplexere Analysen und operative Antwortzeiten nicht mehr als ausreichend bewertete. InterSystems IRIS wurde dort als Datenzugriffs- und Verarbeitungsschicht zwischen bestehender Datenhaltung und produktiven Anwendungen eingeführt. Damit entstand kein vollständiger Neubau der Landschaft, sondern ein Architekturpfad, der bestehende Systeme weiter nutzbar machte und zugleich auf höhere Anforderungen an Geschwindigkeit, Parallelität und Datenverarbeitung reagierte.

Im Ergebnis konnten historische und aktuelle Transaktionen mit hoher Verarbeitungsgeschwindigkeit bereitgestellt, Antwortzeiten deutlich reduziert und mehr als hundert gleichzeitige Verbindungen unterstützt werden. Aussagekräftig ist an diesem Beispiel weniger der Einzelwert als die architektonische Reaktion auf die gewachsene Last. Statt die bestehende Umgebung vollständig zu ersetzen, wurde eine Struktur geschaffen, die Echtzeitverarbeitung, analytische Anforderungen und steigende Nutzungsintensität in einem gemeinsamen Betriebsmodell zusammenführt.

4.2 Hochverfügbarkeit und Ausfallsicherheit technisch eingeordnet

Für geschäftskritische Plattformen ist Verfügbarkeit kein optionales Merkmal, sondern Bestandteil des Betriebsmodells. Sobald Anwendungen operative Prozesse, regulatorische Anforderungen oder kundennahe Services tragen, reicht es nicht aus, nur auf Leistung und Skalierung zu schauen. Dann stellt sich zusätzlich die Frage, wie eine Plattform auf Ausfälle reagiert, wie schnell sie einen Fehlerfall erkennt und wie sich Umschaltung und Recovery so organisieren lassen, dass der Betrieb mit minimaler Unterbrechung weiterlaufen kann. Hier wird aus einer technischen Eigenschaft eine Managementanforderung. Ausfallsicherheit entscheidet unmittelbar über Betriebsrisiko, Servicequalität und wirtschaftliche Folgekosten.

InterSystems IRIS setzt hierfür auf Mirroring als zentrales Hochverfügbarkeitsmodell. Der Grundgedanke dabei ist, dass ein gespiegelter Datenbestand auf einem zweiten System bereitsteht und im Fehlerfall automatisch übernommen werden kann. Aus Sicht des

InterSystems IRIS setzt auf Mirroring als zentrales Hochverfügbarkeitsmodell. Der Grundgedanke dabei ist, dass ein gespiegelter Datenbestand auf einem zweiten System bereitsteht und im Fehlerfall automatisch übernommen werden kann.

Betriebs reicht es nicht, dass ein zweites System vorhanden ist. Wichtig ist, dass Erkennung, Entscheidungslogik und Umschaltung sauber zusammenspielen. Erst dann wird aus Redundanz echte Ausfallsicherheit.

Hochverfügbarkeit lässt sich deshalb nicht sinnvoll über ausschließlich einen Prozentwert beschreiben. Auch zwei redundante Knoten allein garantieren noch keine belastbare Verfügbarkeit. Ob Verfügbarkeit im Betrieb tatsächlich trägt, entscheidet sich vor allem an drei Punkten:

- **Fehlererkennung:** Ein Ausfall muss zuverlässig und schnell erkannt werden.
- **Umschaltung:** Es muss klar geregelt sein, welches System im Fehlerfall übernimmt und wie der Übergang erfolgt.
- **Wiederanbindung:** Anwendungen und nachgelagerte Systeme müssen kontrolliert auf den aktiven Knoten zurückgeführt werden.

In diesem Zusammenhang spielen zusätzliche Komponenten eine wichtige Rolle. Der Arbiter etwa, der als unabhängiger Beobachter den Status aller beteiligten Nodes überprüft und für den Switch zuständig ist. Als Instanz zur Entscheidungsfindung im Fehlerfall stellt er sicher, dass Umschaltungen konsistent und ohne Konflikte erfolgen. In hochverfügbaren Umgebungen wird ein solches Ziel häufig auch in der Kurzform der „**4 Nines**“ beschrieben, also 99,99 Prozent Verfügbarkeit. Ausschlaggebend ist jedoch nicht die Kennzahl allein, sondern die Frage, wie sie im Betrieb technisch erreicht wird.

Was bedeuten „4 Nines“?

„4 Nines“ steht für **99,99 Prozent Verfügbarkeit**. Rechnerisch entspricht das rund **52,56 Minuten ungeplanter Ausfallzeit pro Jahr** oder rund **4,32 Minuten in einem 30-Tage-Monat**. Die Zahl hilft, eine abstrakte Prozentangabe in eine betriebliche Größenordnung zu übersetzen.

$$\begin{aligned} \text{„4 Nines“} &= \text{99,99 Prozent Verfügbarkeit} \\ &= \\ \text{52,56 Minuten ungeplanter Ausfallzeit pro Jahr} \\ &= \\ \text{4,32 Minuten in einem 30-Tage-Monat} \end{aligned}$$

Für die Architekturentscheidung reicht diese Kennzahl allein jedoch nicht aus. Sie sagt noch nichts darüber aus, wie schnell ein Ausfall erkannt wird, wie Umschaltung und Wiederanlauf organisiert sind oder wie sich Anwendungen und nachgelagerte Systeme im Recovery-Fall verhalten. Verfügbarkeit ist deshalb nicht nur eine Kennzahl, sondern das Ergebnis einer konkret geplanten Failover- und Recovery-Logik.

Eine Plattform ist belastbar, wenn sie Skalierung und Ausfallsicherheit in einem Betriebsmodell zusammenführt.

Ein wesentlicher Vorteil dieses Modells liegt darin, dass Hochverfügbarkeit nicht an gemeinsam genutzten Speicher gebunden ist. Damit entfällt ein typischer Single Point of Failure, der in anderen Architekturen durch Shared Storage entstehen kann. Für den Weiterbetrieb bedeutet das: Neben der Datenintegrität wird auch die technische Unabhängigkeit der beteiligten Systeme Teil des Verfügbarkeitskonzepts.

Hochverfügbarkeit bleibt dabei nicht auf klassische Rechenzentrumsmodelle beschränkt. Auch in virtualisierten, cloudbasierten oder Kubernetes-orientierten Umgebungen stellt sich dieselbe Grundfrage: Wie lässt sich Verfügbarkeit so organisieren, dass Redundanz, Umschaltung und Betriebsstabilität zusammenpassen? Gerade in hybriden und verteilten Szenarien wird deshalb wichtig, dass Ausfallsicherheit als Baustein eines gesamten Betriebsmodells gedacht wird und nicht als nachträglich angesetztes Zusatzfeature. Das gilt auch für Cross-Region-Architekturen, in denen Redundanz nicht nur innerhalb eines Rechenzentrums, sondern über geografisch getrennte Standorte hinweg organisiert werden muss. Diese werden häufig angewendet, um regulatorische Anforderungen zu erfüllen oder Ausfälle ganzer Infrastrukturregionen abzufangen.

Stabilität und Skalierung berühren sich an dieser Stelle unmittelbar. Im operativen Betrieb treten beide Anforderungen selten getrennt auf. Systeme mit wachsender Last müssen häufig zugleich hochverfügbar bleiben. Eine Plattform ist daher nicht dann belastbar, wenn sie nur skaliert oder nur ausfallsicher ist, sondern wenn sie beides in einem konsistenten Betriebsmodell zusammenführen kann.





5 Interoperabilität und Legacy-Strategie

Architekturentscheidungen scheitern meistens nicht an neuen Anforderungen, sondern an der vorhandenen IT-Landschaft. Fachanwendungen, Schnittstellen, Datenquellen und historisch gewachsene Abhängigkeiten bleiben über Jahre hinweg Teil des operativen Kerns. Der Wert einer Plattform bestimmt sich daher nicht nur daran, was sie neu aufbauen kann, sondern auch daran, wie gut sie mit Heterogenität umgeht, ohne selbst zur nächsten Komplexitätsschicht zu werden. Für Unternehmen ist das der Praxistest jeder Architektur. Sie muss nicht nur im Zielbild überzeugend sein, sondern auch mit der real existierenden Systemlandschaft arbeiten können.

InterSystems versetzt Unternehmen in die Lage, bestehende Daten und Anwendungen anzubinden und vorhandene Investitionen weiter zu nutzen.

Interoperabilität ist in diesem Zusammenhang kein technisches Zusatzmerkmal. Sie wird zur strategischen Fähigkeit, sobald Integration nicht mehr Ausnahme, sondern Dauerzustand ist. Je mehr Systeme dauerhaft zusammenspielen müssen, desto wichtiger wird ein Integrationsansatz, der bestehende Systeme verlässlich verbindet und neue Anbindungen in eine wiederverwendbare Struktur einordnet. Zum entscheidenden Faktor wird dann, ob sich unterschiedliche Anwendungen, Formate und Prozesse in ein belastbares Integrationsmodell einfügen lassen, ohne dass mit jeder Erweiterung neue Sonderlogiken und zusätzliche Wartungspfade entstehen. Die Managementfrage lautet damit nicht mehr nur „Was können wir anbinden?“, sondern „Wie lange bleibt diese Integrationslogik auch unter wachsender Dichte beherrschbar?“

InterSystems IRIS setzt genau an dieser Stelle an. Die Plattform ist darauf ausgelegt, Datenmanagement, Integrationslogik und Prozessverarbeitung so zusammenzuführen, dass heterogene Landschaften langfristig steuerbar bleiben. InterSystems versetzt Unternehmen damit in die Lage, bestehende Daten und Anwendungen anzubinden und vorhandene Investitionen weiter zu nutzen. Im Mittelpunkt steht nicht der Ersatz bestehender Systeme, sondern ihre strukturierte Einbindung in ein Integrationsmodell, das auch bei wachsender Komplexität tragfähig bleibt. Dieser Punkt ist für die Legacy-Frage entscheidend. Nicht jeder Bestand muss sofort ersetzt werden, aber jede Einbindung sollte in eine belastbare Struktur überführt werden.

5.1 Integration heterogener Systeme

Heterogene Landschaften sind in Unternehmen der Normalfall. ERP- und CRM-Systeme, Fachanwendungen, Datenbanken, APIs, Dateischnittstellen und externe Plattformen müssen parallel integriert werden. Solange nur vereinzelt neue Anbindungen hinzukommen und diese jeweils klar abgegrenzte Anforderungen erfüllen, lassen sich Integrationsaufgaben oft noch mit überschaubarem Aufwand lösen. Die entstehenden Verbindungen bleiben dann in ihrer Zahl begrenzt, ihre Logik ist meist noch nachvollziehbar, und Betrieb sowie Fehlersuche lassen sich im Einzelfall organisieren. Kritisch wird es dort, wo nicht mehr einzelne Integrationen umgesetzt werden, sondern die Anforderung entsteht, fortlaufend neue Systeme, Datenquellen und Prozesse anbinden zu müssen. Dann wächst nicht nur die Zahl der Schnittstellen, sondern auch die Zahl der Sonderfälle, Ausnahmen und dauerhaft zu pflegenden Abhängigkeiten.

Für Unternehmen ist der eigentliche Kipppunkt nicht die Anzahl der Systeme, sondern die fehlende Wiederverwendbarkeit der Integrationslogik.

Die Vielfalt solcher Landschaften ist dabei nicht automatisch ein Problem. Besonders kritisch wird sie aber dort, wo jede zusätzliche Anbindung wieder bei null beginnt: mit eigener Logik, eigener Transformation, eigenem Betriebsaufwand und eigener Fehlersuche. Dann entsteht kein belastbares Integrationsmodell, sondern ein Netz aus Punkt-zu-Punkt-Beziehungen, das mit jeder Erweiterung schwerer zu steuern ist. Für Unternehmen ist das der eigentliche Kipppunkt. Nicht die Anzahl der Systeme allein wird zum Problem, sondern die fehlende Wiederverwendbarkeit der Integrationslogik.

InterSystems IRIS zielt deshalb nicht auf Einzelanbindungen, sondern auf einen Integrationsansatz, der Wiederverwendung und Standardisierung begünstigt. Die Plattform stellt dafür eine strukturierte Integrationsumgebung bereit, in der Systeme verbunden, Nachrichten transformiert und Verarbeitungslogiken geordnet abgebildet werden können. Gleichzeitig stehen Werkzeuge zur Entwicklung, Verwaltung und Überwachung dieser Integrationsstrecken zur Verfügung.

Im laufenden Betrieb manifestiert sich das vor allem in drei Punkten:

- **Wiederverwendbarkeit steigt:** Integrationslogik lässt sich häufiger als Baustein statt als Einzelfall aufbauen.
- **Sonderlösungen nehmen ab:** Abweichungen müssen nicht jedes Mal in einem neuen Integrationspfad münden.
- **Wartungsaufwand wächst langsamer:** Mit steigender Zahl angebundener Systeme nimmt die strukturelle Komplexität nicht im gleichen Maß zu.

Heterogenität verschwindet damit nicht. Unterschiedliche Datenmodelle, Protokolle und Altsysteme bleiben möglich. Der Unterschied liegt in der Struktur, in der diese Vielfalt aufgefangen wird. Um die Qualität einer Interoperabilitäts-Lösung zu bewerten, ist die maßgebende Frage daher nicht mehr, wie viele Systeme sich grundsätzlich anbinden lassen. Ausschlaggebend ist vielmehr, wie belastbar die Integrationslogik auch bei wachsender Dichte bleibt. Damit wird Interoperabilität aus einer technischen Fähigkeit zu einem Architekturprinzip für dauerhafte Steuerbarkeit.

Um die Qualität einer Interoperabilitäts-Lösung zu bewerten, ist die maßgebende Frage nicht, wie viele Systeme sich anbinden lassen. Sondern wie belastbar die Integrationslogik auch bei wachsender Dichte bleibt.

Woran belastbare Interoperabilität erkennbar wird

Die Zahl der unterstützten Schnittstellen allein sagt wenig über die Tragfähigkeit einer Integrationsarchitektur aus. Entscheidend ist, ob sich weitere Systeme anbinden lassen, ohne dass Komplexität, Abstimmungsaufwand und Betriebsrisiken im gleichen Maß zunehmen.

Drei Diagnosefragen helfen bei der Einordnung:



Entstehen neue Anbindungen überwiegend aus wiederverwendbaren Bausteinen oder jedes Mal als Einzelfall?



Bleiben Betriebs- und Fehlersuchaufwände trotz wachsender Integrationsdichte beherrschbar?

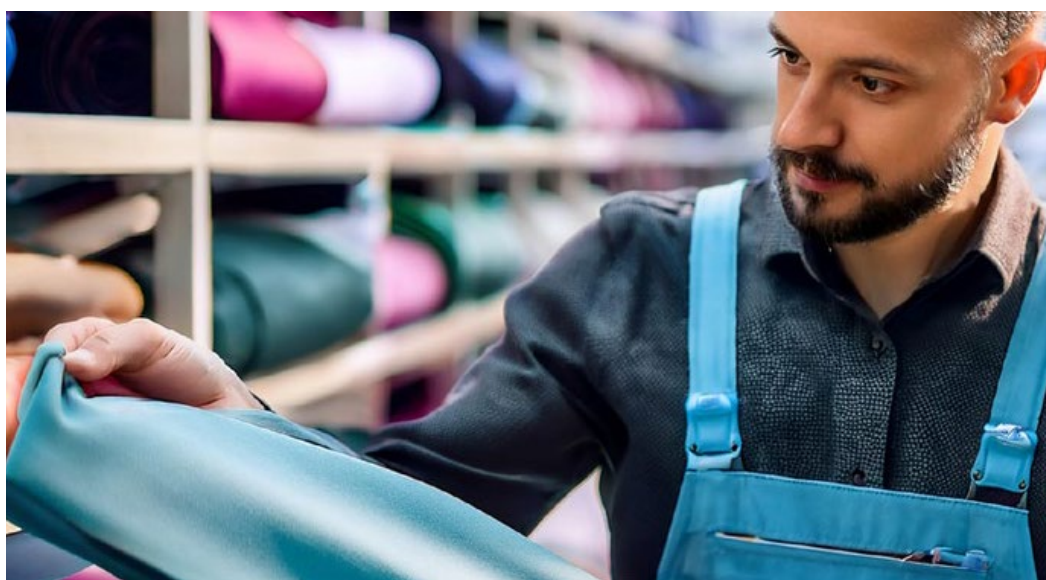


Lässt sich die Rolle eines neu angebundenen Systems klar in die bestehende Integrationslogik einordnen?

Je häufiger diese Fragen mit Ja beantwortet werden können, desto eher liegt ein belastbares Integrationsmodell vor statt einer bloßen Ansammlung technischer Verbindungen.

KUNDENBEISPIEL

Durchgängige Transparenz über Produktion und Energieverbrauch



Der Textilhersteller RÖKONA betreibt eine hochintegrierte Fertigung mit zahlreichen Systemen – von Maschinensteuerung und Betriebsdatenerfassung bis hin zu ERP, EDI und Analytics. Ziel war es, Produktions- und Energiedaten nicht nur zu erfassen, sondern unternehmensweit konsistent nutzbar zu machen.

Dafür baute der InterSystems-Partner update texware auf Basis von InterSystems IRIS eine zentrale Daten- und Integrationsplattform für RÖKONA auf. Diese verbindet operative Systeme, Maschinen- und Verbrauchsdaten sowie analytische Anwendungen in einer gemeinsamen Architektur.

Der entscheidende Mehrwert entsteht durch die Zusammenführung bislang getrennter Datenwelten:

- **Energie- und Ressourcenverbräuche** werden direkt im Kontext der Produktion messbar und auswertbar
- **Daten stehen in Echtzeit** für Monitoring, Steuerung und Optimierung zur Verfügung
- **Neue Anforderungen** lassen sich auf bestehenden Strukturen aufsetzen

Das Ergebnis ist eine durchgängige Transparenz über Produktionsprozesse und Energieeinsatz. Integration wird damit zur Grundlage für Effizienzsteigerung, Nachhaltigkeit und operative Steuerbarkeit.

5.2 Transparenz, Monitoring und Fehleranalyse

Ein bestehendes System wird nicht allein wegen seines Alters zum Problem. Entscheidend ist vielmehr, wie stark es im Laufe der Zeit mit anderen Anwendungen verflochten wurde und wie gut sich diese Abhängigkeiten noch beherrschen lassen. Mit jeder zusätzlichen Integration können Datenflüsse, Zuständigkeiten und Fehlerbilder schwerer nachvollziehbar werden. In verteilten Landschaften führt die Fehlersuche dann häufig über mehrere Werkzeuge, Teams und Zeiträume hinweg. Relevante Informationen liegen in Logs, Einzelsystemen, proprietären Oberflächen und nicht selten im Erfahrungswissen einzelner Personen. Genau dadurch wird Legacy nicht nur technisch, sondern auch organisatorisch schwer steuerbar. Kritisch wird es besonders dort, wo nicht mehr transparent ist, wie Subsysteme in die Gesamtarchitektur eingebunden sind, welche Abhängigkeiten bestehen und welche Folgen Änderungen oder Ausfälle nach sich ziehen.

Transparenz wird damit zur Voraussetzung für jede sinnvolle Legacy-Strategie. Erst wenn Datenflüsse, Prozesszustände und Fehlermuster nachvollziehbar sind, lässt sich bewerten, ob ein System weiter integriert, schrittweise entkoppelt oder ersetzt werden sollte. Hier verschiebt sich die Perspektive von der Technik- zur Managementfrage. Ohne Transparenz lässt sich Legacy nicht steuern, sondern nur verwalten.

InterSystems IRIS adressiert diesen Punkt, indem Integrationsflüsse, Prozessschritte und Fehlerzustände im Zusammenhang sichtbar bleiben. Der Message Viewer zeigt gesendete oder wartende Nachrichten an, und Visual Trace macht den Pfad zusammengehöriger Nachrichten zwischen Business Hosts nachvollziehbar. Über das Management Portal lassen sich außerdem sogenannte Productions verwalten und überwachen. Das sind in InterSystems IRIS die konfigurierbaren Laufzeitumgebungen, in denen Integrationsflüsse, Geschäftslogik und Schnittstellenverbindungen als zusammenhängende Einheit betrieben werden. Gerade in gewachsenen Umgebungen wird dadurch nicht nur die Fehlersuche präziser, sondern auch die Steuerbarkeit der bestehenden Systeme und Schnittstellen verbessert.

Transparenz wird zur Voraussetzung für jede sinnvolle Legacy-Strategie. InterSystems IRIS adressiert diesen Punkt, indem Integrationsflüsse, Prozessschritte und Fehlerzustände im Zusammenhang sichtbar bleiben.

Der entscheidende Mehrwert liegt dabei darin, dass sich ein Prozesspfad über Systemgrenzen hinweg in seinem Zusammenhang nachvollziehen lässt. Erst dadurch wird aus verstreuten Einzelinformationen eine belastbare Grundlage für Fehlersuche und Architekturentscheidungen. Wo Integrationslogik, Nachrichtenpfade und Fehlerzustände sichtbar bleiben, lässt sich präziser eingrenzen, an welcher Stelle ein Problem entsteht und welche Systeme tatsächlich betroffen sind. Für Architekturbeteiligte ist das mehr als ein Betriebsdetail. Es schafft die Voraussetzung, Modernisierungsschritte auf Basis belastbarer Informationen zu priorisieren.

Für die Praxis ergeben sich daraus drei direkte Effekte:

- **Fehlersuche wird zielgerichteter:** Nachricht, Ursache und Prozesspfad lassen sich im Zusammenhang statt nur isoliert betrachten.
- **Datenflüsse bleiben nachvollziehbar:** Integrationslogik verschwindet nicht in verteilten Einzellösungen.
- **Legacy wird steuerbarer:** Systeme können länger sinnvoll eingebunden bleiben, weil ihre Rolle und ihre Abhängigkeiten transparenter werden.

Wo Transparenz vorhanden ist, lassen sich Integration, Entkopplung und Ersatz deutlich nüchterner bewerten.

Gerade der letzte Punkt ist für die Architekturentscheidung zentral. Legacy-Strategie bedeutet nicht automatisch Ablösung. In vielen Fällen ist es sinnvoller, ein bestehendes System zunächst kontrolliert einzubinden, seine Schnittstellen zu stabilisieren und seine Rolle in der Gesamtarchitektur transparenter zu machen, statt es sofort zu ersetzen. Ein belastbares Entscheidungskriterium ist daher weniger das reine Alter eines Systems, sondern: Wie kritisch ist das System für den Betrieb? Wie transparent sind seine Datenflüsse? Wie hoch ist der Änderungsdruck? Und wie groß wäre das Risiko eines Ersatzes? Wo Transparenz fehlt, wird eine Ablösung schnell selbst zum Hochrisikoprojekt. Wo Transparenz vorhanden ist, lassen sich Integration, Entkopplung und Ersatz deutlich nüchterner bewerten.

Was eine sinnvolle Legacy-Strategie auszeichnet

Legacy-Strategie bedeutet nicht automatisch Ersatz. In vielen Fällen ist es strategisch sinnvoller, ein bestehendes System zunächst transparenter, stabiler und besser eingebunden zu machen, bevor über Ablösung entschieden wird.

Vier Fragen sind dafür besonders relevant:

-  **Wie kritisch ist das System für den laufenden Betrieb?**
-  **Wie transparent sind seine Datenflüsse und Abhängigkeiten?**
-  **Wie hoch ist der fachliche oder regulatorische Änderungsdruck?**
-  **Wie groß wäre das Risiko eines Ersatzes im Verhältnis zum Nutzen?**

Erst wenn diese Fragen belastbar beantwortet werden können, wird aus Legacy-Umgang eine steuerbare Modernisierungsstrategie.

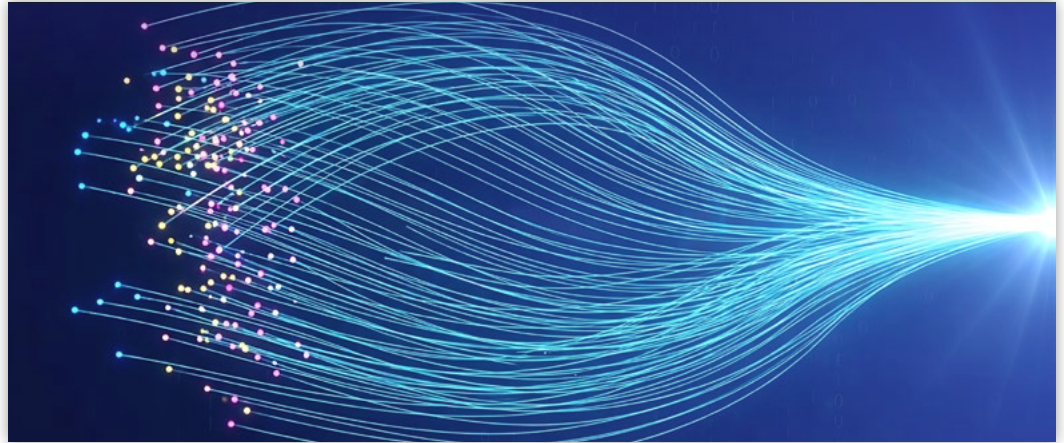
Interoperabilität als Modernisierungspfad



Les Voies Navigables de France stand vor der Aufgabe, eine siloartige Informationslandschaft mit mehreren Datenbanken und Anwendungen neu zu ordnen. Für den Austausch mit Partnern und externen Stellen mussten bislang immer wieder neue Anwendungen gebaut werden. Die Folge war ein schwer wartbares System mit geringer Agilität. Im Zuge einer Modernisierung wurden Daten in zentrale Repositories überführt. Eine Interoperabilitätsplattform sollte diese Repositories nun mit bestehenden Datenbanken und Anwendungen verbinden. InterSystems IRIS wurde dabei als Grundlage für den Data Hub eingesetzt.

Die vorhandene Landschaft wurde dabei nicht in einem Schritt ersetzt. Stattdessen entstand eine zentrale, interoperable Struktur, über die Informationsaustausch, Wartbarkeit und langfristige Steuerbarkeit verbessert wurden. Gerade für den Legacy-Umgang ist das der ausschlaggebende Punkt: Bestehende Systeme bleiben zunächst Teil der Gesamtarchitektur, werden aber über eine klarere Integrationslogik beherrschbarer.

Fazit



Ein Plattformansatz ist erforderlich, wenn Komplexität nicht weiter mit zusätzlicher Komplexität beantwortet werden soll

Ein Plattformansatz ist kein Nice-to-Have und nicht einfach nur die Entscheidung für ein weiteres Werkzeug. Er ist eine strategische und strukturelle Notwendigkeit, um die Chancen einer zunehmend digitalisierten Zukunft zu nutzen.

In dem Moment, in dem neue Anforderungen regelmäßig zusätzliche Schnittstellen, Datenkopien und Abstimmungsaufwand erzeugen, hat sich die Rolle der Architektur bereits verändert. Sie ist dann nicht mehr Enabler, sondern limitierender Faktor. Der Steuerungsaufwand wächst nicht mehr kontrolliert, sondern exponentiell über Übergänge, Zuständigkeiten und Systemgrenzen hinweg.

Genau hier verschiebt sich der Bewertungsmaßstab. Nicht mehr die Leistungsfähigkeit einzelner Werkzeuge ist entscheidend, sondern die Frage, ob die Gesamtarchitektur unter steigender Integrationsdichte, wachsendem Datenvolumen und zunehmendem Betriebsdruck noch steuerbar bleibt.

InterSystems IRIS adressiert diesen Punkt auf architektonischer Ebene. Durch die Zusammenführung von Datenmanagement, Integration, Ausführungslogik und Betrieb entsteht ein gemeinsamer Kern, auf dem neue Anforderungen nicht als zusätzliche technische Strecken umgesetzt werden müssen, sondern als Erweiterung eines bestehenden Zusammenhangs.

Der Unterschied ist grundlegend. Während ein wachsender Technologie-Stack Komplexität weiter verteilt, begrenzt ein plattformbasierter Ansatz die Zahl der Übergänge, an denen Komplexität entsteht. Damit verschiebt sich nicht nur die technische Struktur, sondern auch die wirtschaftliche Logik der Gesamtarchitektur.

Für Unternehmen ergibt sich daraus eine klare Managementfrage: Wie viel zusätzliche Komplexität ist noch vertretbar, bevor sie zur strukturellen Belastung wird?

Diese Frage gewinnt durch die zunehmende Verbreitung von Künstlicher Intelligenz (KI) zusätzlich an Dringlichkeit. KI-Initiativen, ob Automatisierung, Assistenzsysteme oder datengetriebene Entscheidungsunterstützung, setzen voraus, dass Daten konsistent, aktuell und ohne Medienbrüche verfügbar sind. In gewachsenen Landschaften mit verteilten Datenkopien, heterogenen Schnittstellen und unklaren Zuständigkeiten scheitern solche Vorhaben häufig an der darunterliegenden Architektur. Eine Plattform, die Datenhaltung, Integration und Verarbeitung in einem gemeinsamen Kern zusammenführt, ist deshalb nicht nur eine Antwort auf heutige Betriebsprobleme – sie ist die strukturelle Voraussetzung dafür, dass KI im Unternehmen überhaupt belastbar wirksam werden kann.

Warum InterSystems?

InterSystems, ein innovativer Anbieter von Datentechnologie, bietet Kunden in mehr als 80 Ländern aus den Bereichen Gesundheitswesen, Finanzwesen sowie Fertigung und Lieferkette eine einheitliche Grundlage für die Entwicklung von Anwendungen der nächsten Generation.

Unsere cloud-first Datenplattformen helfen Organisationen weltweit, Herausforderungen rund um Interoperabilität, Geschwindigkeit und Skalierbarkeit zu lösen. So können sie das Potenzial ihrer Daten erschließen und Daten auf neue, kreative Weise interpretieren und nutzen.

Seit seiner Gründung im Jahr 1978 steht InterSystems für höchste Servicequalität und unterstützt Kunden und Partner weltweit mit ausgezeichnetem 24/7 Support. InterSystems ist privat geführt, hat seinen Hauptsitz in Boston, Massachusetts, und ist mit 38 Niederlassungen in 28 Ländern weltweit vertreten.

Für weitere Informationen besuchen Sie bitte www.InterSystems.de

InterSystems, A Creative Data Technology Provider.

Rechtliche Hinweise und Markeninformationen

Produkte und Dienstleistungen unterliegen den Allgemeinen Geschäftsbedingungen von InterSystems. Die bereitgestellten Informationen dienen ausschließlich allgemeinen Informationszwecken, können jederzeit ohne vorherige Ankündigung geändert werden und erfolgen ohne Gewähr. Verfügbarkeit, Funktionalität und regulatorischer Status von Produkten können je nach Angebot, Rechtsraum und vorgesehenem Verwendungszweck variieren. Informationen zu bestimmten Angeboten, einschließlich ihrer Verfügbarkeit und ihres regulatorischen Status, erhalten Sie bei Ihrer InterSystems-Ansprechpartnerin bzw. Ihrem InterSystems-Ansprechpartner. Informationen zum Umgang von InterSystems mit Künstlicher Intelligenz finden Sie unter www.intersystems.com/ai-use-guidelines.

© 2026 InterSystems Corporation. Alle Rechte vorbehalten.

InterSystems®, InterSystems HealthShare®, InterSystems IRIS® und Health Connect Cloud® sind eingetragene Marken der InterSystems Corporation. InterSystems IRIS for Health™, InterSystems IntelliCare™ und TrakCare sind Marken der InterSystems Corporation.