

Reinventing OEM Captive Finance on SAP

How leasing providers and OEM captives scale XaaS, modernize leasing, and keep financial control—without breaking the ERP backbone



Agenda

<u>Executive Summary</u>	<u>3</u>
<u>How to Use This Ebook</u>	<u>4</u>
<u>Chapter 1: The Shift From Financing Products to Delivering Outcomes</u>	<u>5</u>
<u>Chapter 2: Why Traditional Leasing Setups Hit a Wall</u>	<u>6</u>
<u>Chapter 3: The Best-in-Class Capability Model</u>	<u>7</u>
<u>Chapter 4: Contract Lifecycle Mastery</u>	<u>8</u>
<u>Chapter 5: 360° Asset Lifecycle Management</u>	<u>9</u>
<u>Chapter 6: Asset based Funding</u>	<u>10</u>
<u>Chapter 7: Finance & Subledger Excellence</u>	<u>11</u>
<u>Chapter 8: Accounting & Control in Hybrid Lease + Service Portfolios</u>	<u>12</u>
<u>Chapter 9: SAP-Centric Target Architecture: Extend Without Breaking the Core</u>	<u>14</u>
<u>Chapter 10: The Operating Model in Real Terms</u>	<u>16</u>
<u>Chapter 11: Measurement That Executives Actually Use</u>	<u>17</u>
<u>Chapter 12: Customer Transformation Story</u>	<u>18</u>
<u>Chapter 13: Results</u>	<u>20</u>
<u>Chapter 14: Common Pitfalls</u>	<u>21</u>
<u>Chapter 15: Roadmap — How to Start Without Disruption</u>	<u>22</u>

Executive Summary

Leasing and OEM captive finance are entering a period of structural change. For decades, many portfolios were built around stable asset ownership models, predictable contracts, and periodic billing. Those fundamentals are being reshaped by a different customer expectation: not ownership, but **outcomes**. Customers want availability, service performance, predictable costs, and flexibility to adjust as their business changes. That demand is accelerating the move toward subscription, pay per use, and bundled offerings often described as XaaS.

For OEM captives and leasing providers, this shift creates an opportunity to grow recurring revenue, increase customer lifetime value, and differentiate through service integration. Yet it also introduces complexity that traditional leasing setups were not designed to handle. Contracts change more frequently. Assets move, get swapped, upgraded, refurbished, redeployed, and returned. Billing must reflect usage, service levels, and changing scope. Finance teams need audit ready traceability across a growing volume of events. And funding becomes more strategic as balance sheet intensity increases.

SAP-based organizations face a particular challenge: SAP ERP is often the backbone for finance, controlling, procurement, and group reporting. Transformation cannot come at the cost of losing financial integrity or introducing uncontrolled interfaces. Successful organizations therefore modernize with a guiding principle: **extend what must be extended, integrate what must be integrated, and keep governance firm.**

In practice, best-in-class organizations integrate four dimensions that have historically been fragmented:

1. **Contract lifecycle mastery** that supports frequent changes without breaking control.
2. **360° asset lifecycle management** that connects operational truth to commercial and financial truth.
3. **Finance and subledger excellence** that automates postings and ensures auditability at scale.
4. **Asset based Funding capabilities** that connect portfolio growth to refinancing and profitability.

This ebook provides a practical, SAP-centric blueprint for that transformation. It explains the market dynamics, the operational fault lines, and the “target state” capabilities that enable scale. It also includes one real-world transformation example of a global equipment and document solutions provider that reinvented its commercial model toward XaaS and modern leasing while maintaining financial control and improving operational scalability.

What you will take away: a narrative that executives can align on, plus concrete guidance for architects and finance leaders who must translate strategy into a stable, scalable SAP-based operating model.

“Modern leasing is no longer a contract business. It’s a contract–asset–finance system.”

How to Use This Ebook

This ebook is written for decision-makers and experts who are responsible for turning a leasing and XaaS strategy into operational reality.

If you lead an OEM captive, your priorities likely include growing attach rates, improving portfolio profitability, and enabling new commercial models that strengthen customer relationships. At the same time, you must manage risk: credit exposure, asset residual values, compliance, auditability, and the cost of funds. You might also be working across multiple regions, each with different process maturity and regulatory requirements.

If you lead a leasing provider, you might be expanding beyond classic financial leases into rental, subscriptions, and service bundles. You may be balancing legacy portfolios with new offerings and trying to harmonize systems and processes that evolved over years. You may also be under pressure to reduce manual work, accelerate close, improve billing accuracy, and gain end-to-end transparency.

Finally, if you are an IT or SAP stakeholder, you are likely navigating a difficult equation: deliver agility for the business, keep SAP clean and stable, reduce complexity, and ensure integration is secure, auditable, and scalable.



Chapter 1:

The Shift From Financing Products to Delivering Outcomes

Why it matters: Customers want availability, performance, and predictable cost—not asset ownership. That shift turns financing into a continuous lifecycle responsibility rather than a periodic billing process.

What you'll learn: How outcome-based expectations reshape contracts, assets, finance, and funding—and why traditional setups reach their limits.

The move toward XaaS is often described as a new pricing model. In reality, it is a new way of creating and delivering value. Customers are increasingly buying assurance: that critical equipment will be available, maintained, compliant, and cost-transparent. They want the ability to scale up and down. They want service included. They want predictable payments. And they want the provider to take responsibility for the lifecycle complexity that used to sit on the customer side.

For OEMs, this creates strategic leverage. You are no longer competing only on product features; you are competing on lifecycle performance. Captive finance becomes not just a financing instrument but a growth engine—enabling bundled offerings, smoothing adoption friction, and creating long-term relationships supported by service and data.

However, as soon as an OEM captive or leasing provider introduces more flexible and service-inclusive models, the business starts to operate in a fundamentally different pattern. In classic leasing, many events are periodic and predictable: monthly billing, periodic accounting, predefined end-of-term options. In XaaS, events become continuous and frequent: swaps, upgrades, maintenance, contract amendments, usage settlement, and service credits. That event volume changes the operating model.

There is also a financial dimension that cannot be ignored. Retaining asset ownership longer increases balance sheet intensity and makes funding strategy more critical. Profitability becomes more sensitive to utilization, downtime, and refurbishment cycles. Residual value management becomes not only a risk topic but a margin driver. Revenue recognition and cost allocation become more complex as bundles mix lease components with service components.

In short: XaaS and modern leasing require a tighter coupling between contract reality, asset reality, and finance reality than traditional system landscapes usually provide.

“XaaS is an operating model shift. If contracts, assets, finance, and funding remain separate, scale becomes impossible.”



Chapter 2:

Why Traditional Leasing Setups Hit a Wall

Why it matters: Fragmented systems can cope with stable contracts. They fail when amendments, swaps, and hybrid bundles become the norm. Misalignment spreads into billing disputes, manual postings, and funding blind spots.

What you'll learn: The four structural breakpoints that prevent scaling—and how to recognize them early.

Many organizations have SAP ERP at the core and rely on a set of surrounding solutions for contracts, billing, asset tracking, service management, and reporting. Over time, this landscape can become functional but fragmented. Each system captures a slice of truth. The problem emerges when the business model introduces frequent change—because then the cost of misalignment increases dramatically.

A typical breaking point is contract change management. In traditional models, contract amendments are relatively rare and are often processed manually. In modern portfolios, change becomes the norm. Customers request flexibility: add-ons, partial returns, temporary

replacements, pricing adjustments, extensions, early terminations, and bundles that evolve. If the contract lifecycle is not managed as a controlled, event-driven process, changes become slow, inconsistent, and costly.

The second breaking point is the divergence between operational truth and commercial truth. Assets move in the real world. They are delivered, installed, replaced, repaired, returned, refurbished, and redeployed. If the asset lifecycle is tracked in operational tools but not tightly connected to contract objects, billing quickly becomes error-prone. Disputes rise. Credit notes increase. Customer trust erodes. And finance teams spend more time reconciling than analyzing.

The third breaking point is finance scalability. When contract and asset events increase, finance faces a volume problem. Manual postings, spreadsheets, and reconciliations may work in stable environments but become operational risk under high change frequency. Closing cycles extend, audit trails weaken, and the organization loses clarity on portfolio profitability—exactly when it needs clarity most.

Finally, funding becomes a limiting factor. As portfolios scale and asset ownership periods extend, organizations require stronger refinancing structures and better transparency on funding costs. Without clear mapping between assets, contracts, and funding instruments, funding decisions are made with incomplete information. That can lead to higher costs of funds, suboptimal risk management, and constraints on growth.

These issues often present themselves as “system problems,” but they are fundamentally **operating model problems**. Systems only reflect the deeper need: an integrated, event-driven foundation where contracts, assets, finance, and funding are not separate worlds.

“XaaS is an operating model shift. If contracts, assets, finance, and funding remain separate, scale becomes impossible.”



Chapter 3:

The Best-in-Class Capability Model

Why it matters: Scaling modern leasing requires four integrated capabilities that reinforce each other. Without all four, growth is either slow, risky, or expensive.

What you'll learn: A practical model to structure transformation decisions across business and IT.

Best-in-class leasing and OEM captive organizations operate with a capability model that treats the portfolio as an integrated lifecycle system. They do not simply manage contracts. They manage the interplay between commercial agreements, physical asset reality, and financial truth. This capability model can be described in four pillars.

The first pillar is **Contract Lifecycle Mastery**. In modern portfolios, the contract must function as a structured object that can evolve without loss of control. That requires standardization, governance, and automation. It means the organization can launch new offerings without rebuilding the process every time and can handle contract changes quickly while maintaining auditability.

The second pillar is **360° Asset Lifecycle Management**. Assets are the economic engine of leasing and XaaS. They also represent risk. Best-in-class organizations maintain a consistent asset view across acquisition, deployment, operation, swaps, returns, refurbishment, redeployment, and disposal. They connect asset events directly to contract obligations and financial effects.

The third pillar is **Finance & Subledger Excellence**. High change frequency requires event-based automation. Finance needs postings that are triggered by contract and asset events, with clear traceability. This is not only about efficiency. It is about control: the ability to explain every posting and every invoice line back to a contract event and an asset state.

The fourth pillar is **Funding**. Portfolio growth must be linked to refinancing capacity and to clear allocation of funding costs. Best-in-class organizations manage asset pools and funding instruments as part of the operating model, not as an afterthought. That capability supports scale and provides transparency on profitability after funding costs.

Throughout the rest of this ebook, we will dive into each pillar in depth and translate it into practical process and architecture patterns for SAP-based environments.

“Flexibility without governance is cost. Governance without flexibility is lost growth.”



Chapter 4:

Contract Lifecycle Mastery

Why it matters: In modern portfolios, the contract is a living object. Changes must be fast, controlled, and financially explainable—otherwise flexibility collapses into exceptions.

What you'll learn: How to standardize commercial patterns while preserving customer-specific flexibility.

In modern leasing and XaaS, a contract is no longer a static document. It is a lifecycle object that must adapt to customer needs while remaining controlled. That is why “contract lifecycle mastery” matters. Without it, flexibility becomes chaos: every customer change creates manual work, exceptions increase, and downstream processes break.

Contract lifecycle mastery begins with standardization. The goal is not to reduce flexibility, but to structure it. Leading organizations define modular contract patterns that support common scenarios: standard lease, lease with service, subscription bundles, usage-based add-ons, mid-term upgrades, and end-of-term options.

When these patterns are standardized, contract creation becomes faster, approvals become simpler, and downstream automation becomes possible.

The second element is controlled change management. In a mature operating model, every contract change is treated as an event with defined rules. An upgrade may require asset swap processes, billing adjustments, and recalculated schedules. A partial return may require reallocation of bundle components and updated revenue/cost assumptions. An extension may trigger changed end-of-term logic and funding implications. The key is that changes are not handled as ad-hoc exceptions but as structured events with workflows and traceability.

The third element is integration of commercial and operational terms. Modern bundles combine multiple components: asset usage, maintenance services, consumables, insurance, guarantees, and sometimes performance-based commitments. Best-in-class organizations design contracts so that

each component can be priced, billed, and accounted for appropriately. This does not mean every contract becomes complex. It means complexity is modeled explicitly, rather than hidden in spreadsheets.

A final element is customer experience. Flexibility is a value proposition only if it can be delivered quickly and accurately. When the contract lifecycle is mature, customers can request changes with confidence that billing will reflect reality, service levels will remain intact, and the provider's operations can execute without delay.

Contract lifecycle mastery is therefore the first pillar of scaling modern leasing and XaaS: it transforms flexibility from a promise into an executable capability.

“If contract change becomes an exception, scale becomes impossible.”



Chapter 5:

360° Asset Lifecycle Management

Why it matters: Assets are the economic engine of leasing and XaaS. Utilization, downtime, refurbishment, and residual value determine margin—yet many organizations manage assets in disconnected views.

What you'll learn: How lifecycle transparency reduces cost-to-serve and protects portfolio value.

If the contract is the commercial backbone of leasing and XaaS, the asset is the economic backbone. Asset lifecycle management is where profitability is created or destroyed—through utilization, downtime, maintenance cost, refurbishment cycles, and residual value realization.

In many organizations, asset management is split across functions. Operations may track physical assets and service history, while finance tracks capitalized values and depreciation. Contract systems may track which asset is assigned to which customer. When these perspectives are disconnected, the organization loses clarity. It becomes

difficult to answer basic but crucial questions: Which assets are idle? Which assets are late in return? Which assets are under-maintained? Which asset classes generate the best contribution margin after service and funding costs?

A best-in-class approach establishes a consistent asset record across the lifecycle. That record includes not only identifiers and ownership information but also status, configuration, location, condition, warranty, service history, and contractual linkage. The link between asset and contract is particularly critical because it enables accurate billing and correct financial effects of asset events.

In XaaS models, asset events happen frequently. A customer might need a replacement due to downtime. A device might be upgraded mid-term. A fleet might be scaled down temporarily. Returned assets might be refurbished and redeployed. Each event changes utilization, cost, and revenue patterns. Asset lifecycle management ensures these events are captured and managed as

structured processes rather than as ad-hoc interventions.

Moreover, asset management becomes a strategic lever for sustainability and margin. Refurbishment and redeployment extend lifecycle value and support circular economy strategies. End-of-life pathways affect both cost and brand reputation. When asset lifecycle management is mature and integrated, these decisions can be optimized based on data, not intuition.

In summary, 360° asset management is not a supporting function. It is a core capability that enables profitability, customer experience, and scale—especially in modern leasing and XaaS.

“You can’t optimize what you can’t see—especially not across the asset lifecycle.”



Chapter 6:

Asset based Funding

Why it matters: As asset ownership periods extend, funding becomes a structural constraint—or a growth accelerator. Transparency on allocation and profitability after cost of funds becomes essential.

What you'll learn: How to link assets and contracts to funding structures without losing traceability.

Many transformations treat funding as a downstream topic: something treasury “handles” once the business has grown. In modern leasing and XaaS, that approach is increasingly risky. As asset ownership periods extend and portfolios scale, funding capacity and cost become structural constraints. Without a robust Asset based Funding capability, growth can become expensive or slow.

In a mature model, funding is integrated into portfolio management. Assets and contracts are linked to funding sources such as credit facilities, refinancing lines, or securitization structures. The organization can transparently allocate funding costs to portfolios, enabling profitability analysis after cost of funds. This becomes particularly important when interest rates fluctuate or when the portfolio mix shifts toward longer asset retention.

Asset based Funding capabilities also support governance and risk management. They enable tenor matching, concentration monitoring, and visibility into liquidity needs. They support reporting requirements and facilitate more professional interactions with banks, investors, and rating stakeholders. For OEM captives, this can be a strategic advantage: it enables the captive to scale business programs and support sales growth without placing undue strain on the balance sheet.

Most importantly, Asset based Funding capability turns capital into an enabler rather than a bottleneck. When funding structures are transparent and integrated, leadership can make growth decisions with confidence, knowing how portfolio changes affect capital requirements and funding costs.

“If funding is a black box, growth becomes expensive by default”



Chapter 7:

Finance & Subledger Excellence

Why it matters: Modern leasing creates high event volume. Manual postings and end-of-month corrections cannot keep up without increasing audit risk and customer friction.

What you'll learn: How event-driven finance creates traceability, speed, and control in SAP-centric environments.

Finance becomes the stress test of any leasing and XaaS transformation. As soon as contract and asset events increase, the finance organization must either automate or drown in reconciliation. But automation in finance is not only about speed; it is about governance, auditability, and the ability to explain outcomes.

In traditional landscapes, finance often relies on periodic processing: monthly postings, batch invoicing, and manual adjustments. That approach struggles when events become frequent. A contract upgrade might require reallocating pricing components. An asset swap might require changing depreciation assumptions. A service credit might require adjustments to revenue allocation. If these changes are handled manually, errors become inevitable.

A best-in-class approach uses event-driven finance logic. The principle is straightforward: operational events that change economic reality must trigger consistent financial effects. When a contract is activated, postings reflect commencement. When an asset is delivered, its financial representation aligns. When a contract is amended, the system updates the relevant schedules. When invoices are generated, they are traceable to contract and asset states. When a return occurs, the asset's status and valuation pathways are captured.

For SAP-based organizations, this typically means maintaining SAP ERP as the system of record for financial reporting while ensuring that subledger and billing processes remain tightly integrated and explainable. A critical success factor is traceability: every posting should be explainable via a chain from financial document to invoice to contract event to asset event. That chain is what enables both internal control and external audit confidence.

Expert-level attention is often required around how different components of a bundle are represented. A modern offering might include an asset usage component plus service components. Some elements behave like lease components, others like service revenue. Organizations need consistent rules for how these are posted, how revenue is recognized, and how costs are allocated. The goal is not to turn the ebook into an accounting manual. The goal is to acknowledge that scaling requires clear financial modeling and consistent integration patterns.

Finance excellence also improves management insight. When postings and billing are event-driven and consistent, the organization can analyze profitability by segment, contract model, and asset class with much higher confidence. That enables decisions on pricing, portfolio steering, and funding strategy—turning finance from a reporting function into a performance function.

“Automation is valuable, but explainability is mandatory.”

Chapter 8:

Accounting & Control in Hybrid Lease

Why it matters: XaaS bundles mix lease components and service components. Without clear accounting logic, organizations lose consistency, increase audit friction, and limit automation.

What you'll learn: How to structure contracts, pricing, and event flows to support consistent treatment of lease vs non-lease components—without turning operations into an accounting exercise.

Innovating commercial models is easy to describe and hard to industrialize. The hardest part is not the headline “subscription” or “usage-based.” The hardest part is what sits underneath: a contract structure that remains operationally manageable and financially consistent when it changes. In SAP-based leasing organizations, accounting and control are not an afterthought; they are the guardrails that enable scale. As portfolios move toward XaaS, those guardrails must become more explicit—because XaaS offerings often combine elements that behave like a lease with elements that behave like a service.

From an IFRS 16 perspective, the core question is whether a contract (or part of a contract) conveys the right to control the use of an identified asset for a period of time in exchange for consideration. Many modern offerings include such a right, even if the commercial packaging is “as a service.” The practical implication is not that every offering must be forced into a single label, but that organizations need a disciplined way to identify and consistently treat lease components and non-lease components within hybrid bundles.

This is where many transformations create friction. Commercial teams understandably want modular bundles and rapid changes. Finance teams require consistent component logic, allocation principles, and audit trails. If this tension is not resolved structurally, the organization ends up with manual

workarounds: spreadsheets that allocate consideration, ad-hoc journal entries at close, and reconciliation between operational billing and accounting treatment. Under scale, those workarounds become expensive and risky.

A robust approach begins with **component clarity**. In hybrid portfolios, a single customer agreement may include an asset usage element (which may be a lease component) and several service elements such as maintenance, consumables, insurance, monitoring, or performance guarantees. The operational need is to price and bill the bundle cleanly. The accounting need is to represent the economics accurately and consistently, including how variable consideration behaves. The transformation requirement is to embed this logic into the contract model so that the correct treatment is not reinvented contract by contract.

“The more innovative the offering, the more disciplined the component logic must be.”

A particularly important discipline is the handling of **lease and non-lease components**. IFRS 16 provides the framework for separating and accounting for lease components and non-lease components, and also allows certain practical expedients in specific cases. In real portfolios, the decision to separate components (and the method used to allocate consideration) must be applied consistently—especially when contracts are amended over time. In other words, component logic must be designed for lifecycle change, not only for contract inception.

That lifecycle aspect becomes critical with mid-term upgrades, swaps, and partial returns. When an asset is replaced, the economics may change: remaining

consideration, service scope, and timing are affected. If the system landscape cannot represent contract modifications as structured events, finance will be forced to “repair” the story at period end. A scalable model treats a modification not as an exception but as a managed change event with clear triggers: what changes in the contract object, what changes in asset assignment, what changes in billing, and what changes in accounting schedules.

Another recurring complexity in XaaS portfolios is **variable consideration**, particularly usage-based pricing. Usage-based charges may be genuinely variable and dependent on future events. From a finance and reporting perspective, it becomes crucial to distinguish between fixed or in-substance fixed payments and variable payments tied to usage or performance. Operationally, this distinction impacts settlement design. Financially, it impacts how the organization explains revenue and how it ensures that billing reflects the appropriate measurement principles. A mature system design therefore treats usage data as a first-class input into settlement, with clear linkage to the contract terms that define how usage becomes billable consideration.

Equally important is the treatment of **service performance obligations** that sit alongside asset usage. Even when the offering is marketed as “all inclusive,” the organization needs to understand what part of the customer value relates to the use of an asset and what part relates to ongoing services. This clarity drives not only accounting treatment but also profitability analysis. Without it, the organization struggles to answer a deceptively simple question: are we making margin because of asset utilization, because of service efficiency, or because of pricing power? Those answers matter for portfolio steering.

For SAP-based organizations, the accounting discussion quickly becomes a system and data discussion. To support IFRS-consistent treatment at scale, the contract model must carry the attributes that finance needs: component classification, allocation logic, modification handling, and traceability to the underlying asset. The billing model must be able to generate invoice lines that can be explained and reconciled to those components. And the subledger model must provide a transparent chain from event to posting so that auditors and internal control functions can follow the logic without relying on tribal knowledge.

The goal is not to turn operations into accountants. The goal is to design contract and event structures so that the accounting outcome becomes a natural consequence of operational truth. When that happens, the organization reduces manual interventions, accelerates close, improves audit readiness, and gains confidence to expand hybrid offerings without losing control.

“The more innovative the offering, the more disciplined the component logic must be.”

From a lessor perspective, the operational challenge is not only contract inception, but contract evolution. Upgrades, swaps, partial returns, extensions, and early terminations are not corner cases in modern portfolios; they are normal operations. Each of these events can change the economics—what is billed, what services are delivered, what asset is used, and how costs flow.

The transformation requirement is therefore to design the contract model for modification. Contract changes must be represented as structured events with explicit rules, not handled as ad-hoc exceptions. When an upgrade occurs, the organization must know what changes in the bundle, what changes in asset assignment, how pricing adjusts, and what downstream settlement and posting effects follow. When a partial return occurs, the system must be able to re-scope the contract and reflect the change without “month-end repair.”

A second complexity is variable consideration, especially usage-based pricing. Usage charges align pricing to customer outcomes, but they demand precise settlement logic: what usage data is relevant, how it is measured, how it maps to contract terms, and how it becomes billable consideration. Mature organizations treat usage as a governed input into settlement and ensure that the relationship between measured usage and contractual billing rule is transparent and explainable.

“When accounting logic is embedded in the contract model, automation becomes audit-ready by design.”

This section provides an operationally oriented perspective on accounting considerations and does not replace formal accounting advice. Organizations should confirm specific interpretations and policy choices with their auditors and accounting advisors.

****Brief lessee context (why your customers care):**** Many of your customers are also under IFRS 16 pressure. Lessees often recognize a right-of-use asset and lease liability for leases, which affects balance sheet presentation, key ratios, and internal approval paths. As a result, customers frequently scrutinize hybrid offerings and ask for clarity: what part relates to asset use versus services, how fixed and variable payments behave, and how modifications are documented. Even when customers seek “as-a-service” simplicity, their accounting and audit stakeholders often demand structured documentation and a consistent modification record. For the lessor, transparency is not merely compliance-friendly—it can accelerate sales cycles and reduce disputes later.

Chapter 9:

SAP-Centric Target Architecture: Extend Without Breaking the Core

Why it matters: SAP ERP is the backbone for governance and reporting. The challenge is enabling modern leasing capabilities around it while keeping integration auditable and stable under scale.

What you'll learn: A product-agnostic layered architecture and a step-by-step integration story from quote to cash to funding.

In SAP-based leasing and captive environments, architecture decisions are rarely neutral. SAP ERP is not just another system; it is the backbone for financial governance, group reporting, controlling discipline, and auditability. That is why modernizing leasing and XaaS capabilities cannot be approached as a simple replacement of one application with another. The objective is to establish a target architecture in which modern contract and asset lifecycle processes can evolve rapidly—while SAP continues to provide the stable financial “ground truth.”

A reliable target architecture therefore starts with clear **system roles** and an explicit **event model**. System roles answer the question: where does each truth live? The event model answers the question: how does reality move through the landscape when something changes?

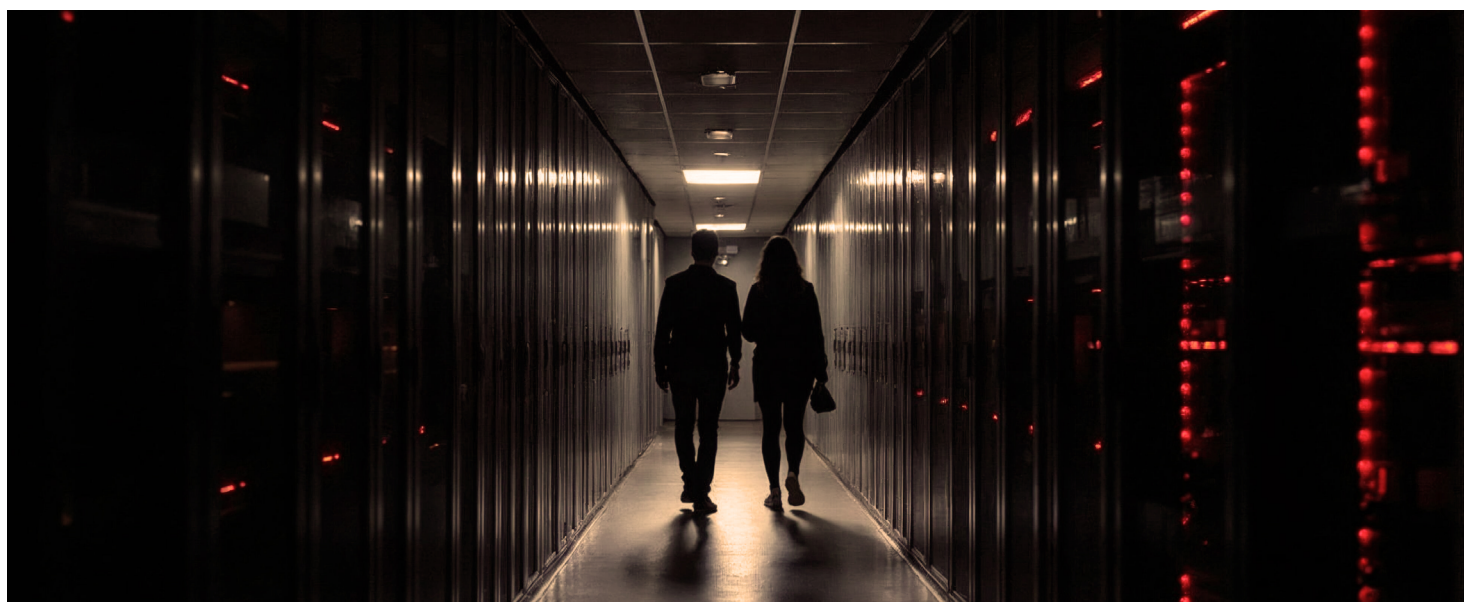
“The target state is one truth—expressed consistently across contract, asset, finance, and funding.”

A proven pattern is a layered model. At the top sits the engagement layer: sales channels, portals, partner channels, and service touchpoints that create commercial intent and capture customer interactions. In the middle sits an operational layer where contract lifecycle management, asset lifecycle event handling, settlement/billing logic, and funding portfolio management are executed. At the bottom sits SAP ERP as the financial core—FI/CO and related functions—consuming structured, explainable postings and enabling statutory and management reporting.

What makes this architecture work is not the layering itself, but the integrity of the integration between layers. In modern leasing and XaaS, the volume of changes is too high for “periodic synchronization” to remain safe. Integration must be designed for event frequency and for control. That typically means four principles.

First, the contract object and the asset object must be linkable through stable identifiers and consistent master data governance. If the organization cannot reliably connect a billing line to a contract state and the assigned asset state, reconciliation becomes permanent. Second, lifecycle changes must be represented as explicit events—activation, swap, upgrade, extension, partial return, termination, return, refurbishment, redeployment. Third, settlement must be designed as a controlled transformation: commercial terms plus measured usage plus lifecycle events produce billable items. And fourth, postings into SAP must be traceable: finance must be able to explain why a posting occurred, which commercial event triggered it, and which asset reality it reflects.

“Integration isn’t a technical detail—it’s the business control system.”



The Integration Story (End-to-End, From Quote to Funding)

To make the target architecture tangible, it helps to follow a single contract through the landscape. The story begins with a quote that defines the commercial bundle: an asset usage element plus service elements and possibly usage-based charges. When the customer accepts, the contract is created as a lifecycle object with defined components and rules. At this point, the architecture must ensure that the contract carries the attributes that later drive billing and accounting: component classification, pricing logic, and event handling rules.

Next comes delivery and activation. An asset is onboarded into the portfolio with its identifier, configuration, and status. The contract is linked to the asset assignment. This linkage is not cosmetic; it is the hinge that connects commercial obligation to physical reality. Activation triggers the initial settlement and the start of the operational lifecycle. The event is recorded so that the system can answer later: when did service begin, what asset was used, and under which terms?

During operation, lifecycle events occur. A swap may be required due to downtime; an upgrade might be requested; a customer might scale down temporarily. Each event changes the relationship between contract and asset and may change billable consideration. The architecture’s job is to ensure that these events are captured once, consistently, and propagated through settlement and finance. If the swap is recorded operationally but not reflected in contract state, billing errors are not a possibility—they are a certainty.

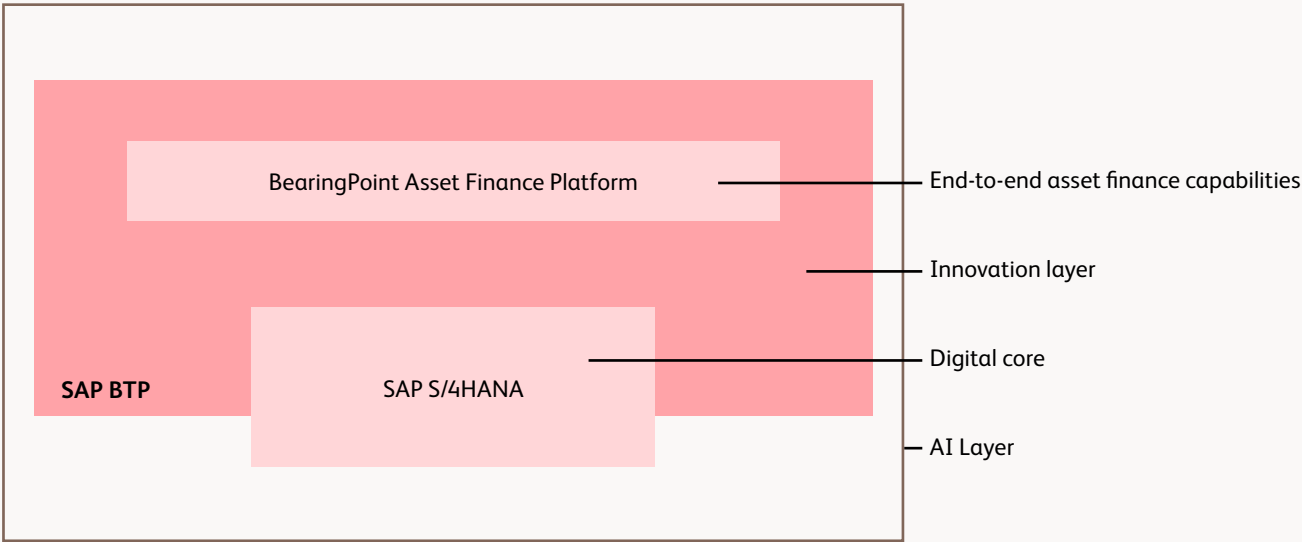
Settlement then becomes the controlled point where contract terms, asset assignment, and usage data are transformed into billable items. In a mature design, settlement logic is transparent: it can show which term produced which charge and which event caused which adjustment. This transparency is essential not only for customer experience but for auditability.

Finally, postings are created and integrated into SAP ERP. The key requirement here is explainability. SAP receives postings that can be traced back to the settlement outputs and further back to contract and asset events.

That traceability is what enables finance to close efficiently and defend outcomes with confidence. Funding allocation can be layered on top by linking contract and asset pools to funding structures, allowing the organization to analyze portfolio profitability after cost of funds and to steer growth with better capital insight.

The architectural message is simple: modern leasing and XaaS do not require a fragile web of interfaces. They require an event-driven integration story where each change is captured as an event, each event has controlled financial implications, and SAP receives postings that are explainable by design.

“Clean core is not a restriction—it’s the stability that makes innovation scalable.”



Chapter 10:

The Operating Model

Why it matters: Architecture alone does not create scale. Without a clear operating model, even the best-designed systems will deteriorate into manual workarounds, inconsistencies, and growing operational risk. In modern leasing and XaaS, success depends on a shared understanding of how events are created, owned, and processed across contract, asset, finance, and funding functions.

What you'll learn: Why unclear event ownership is the hidden root cause of many process failures—and how a well-defined operating model enables stable, scalable automation by aligning responsibilities, data, and governance.

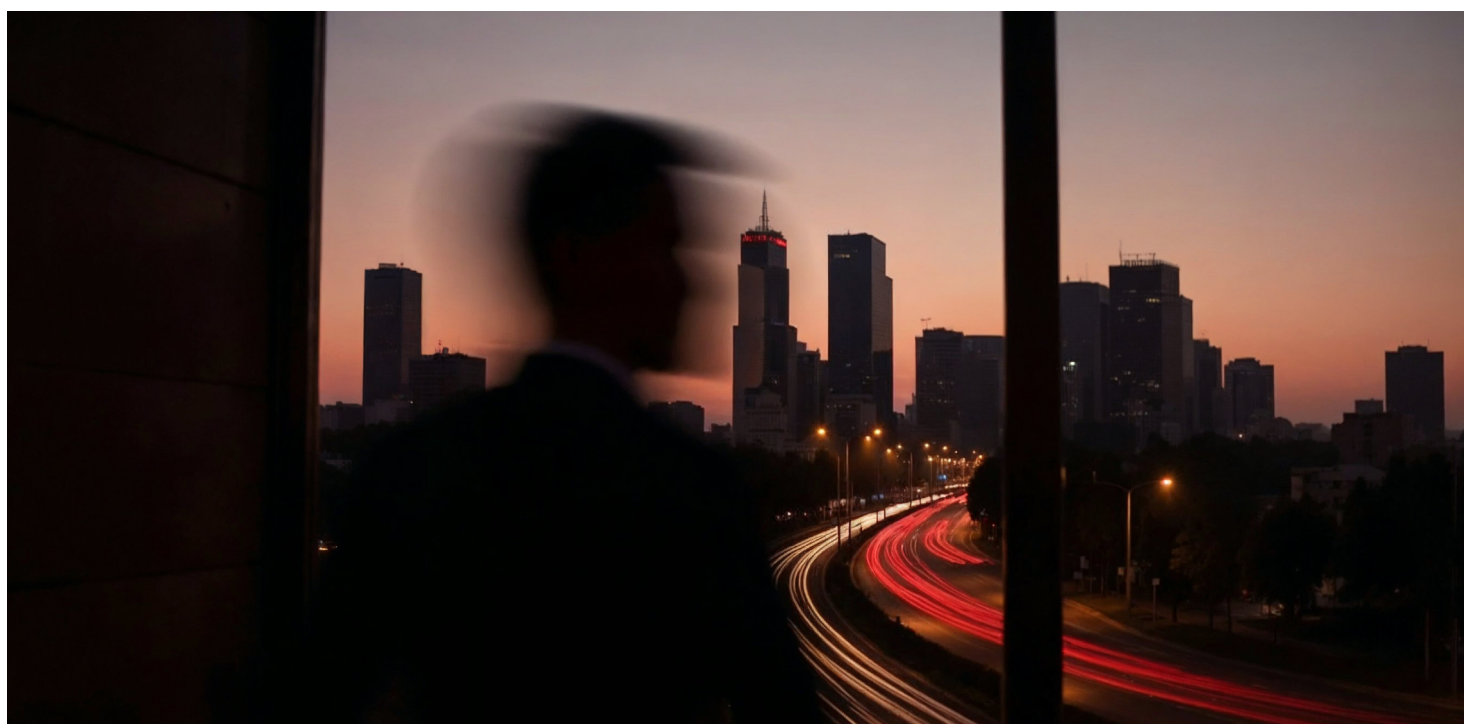
Even the best architecture will fail if the operating model remains fragmented. Modern leasing and XaaS require a shared understanding of responsibilities across contract operations, asset operations, finance operations, and funding operations. The goal is not to create bureaucracy. The goal is to create clarity so that event-driven automation remains correct.

One of the most common root causes of billing disputes is not a system bug, but unclear event ownership. If an asset swap occurs but the contract is not amended correctly, the customer will be billed incorrectly. If a return is processed operationally but not reflected in contract state, billing continues. If service credits are agreed commercially but not represented as structured events, finance resolves the issue manually—again and again.

A mature operating model defines event ownership and process governance. It creates a controlled taxonomy of event types: delivery, activation, swap, upgrade, return, termination, extension, and so on. It defines who triggers each event, which approvals apply, and how it propagates through contract, asset, billing, and finance. It also defines master data responsibilities: who owns asset identifiers, who governs customer master data, who maintains product catalogs and pricing structures.

Crucially, this operating model enables automation to remain stable under growth. Without it, each portfolio expansion introduces new exceptions, and the organization slowly returns to manual workarounds.

“Automation doesn’t fail because of systems—it fails because no one owns the event.”



Chapter 11:

Measurement That Executives Actually Use

Leasing and XaaS transformations can easily become “technology programs” without measurable outcomes. That is why executives need a KPI framework that connects operational capability to commercial performance and financial control.

The most valuable KPI sets are those that show end-to-end performance. For example, contract cycle time matters, but it matters most when connected to billing accuracy and dispute rates. Asset utilization matters, but it matters most when linked to funding cost and margin. Finance automation matters, but it matters most when connected to close speed and audit confidence.

A mature KPI framework typically includes a small number of “north star” measures and a larger number of operational diagnostics. North star measures might include recurring revenue share, net contribution margin after service and funding costs, and customer retention. Operational diagnostics include contract amendment throughput, asset redeployment cycle time, invoice accuracy, manual posting ratio, and close cycle time.

This measurement approach does not only prove value. It supports steering. It enables leadership to identify where the operating model is breaking and where process improvements yield the highest return.



Chapter 12:

Customer Transformation Story

Why it matters: Seeing the journey as a story helps stakeholders align on what actually changes: governance, event ownership, and the link between operational and financial truth.

What you'll learn: The starting point, the design choices, and the outcomes—without relying on brand names or confidential metrics.

A global provider of equipment and document solutions had built a strong market position over decades through product innovation and a large installed base. Its captive organization played an important role in enabling customers to adopt technology through financing programs. Over time, however, the company recognized that customer expectations were shifting. Customers wanted flexible service-inclusive models, predictable monthly cost, and simplified lifecycle responsibility. Competitors were also moving toward subscription-like offerings, increasing pressure on differentiation.

The organization decided to evolve its model toward a combined portfolio of modern leasing and XaaS offerings. Strategically, the logic was clear: more recurring revenue, deeper relationships, more data-driven lifecycle management, and a stronger service proposition. Operationally, the shift exposed structural limitations.

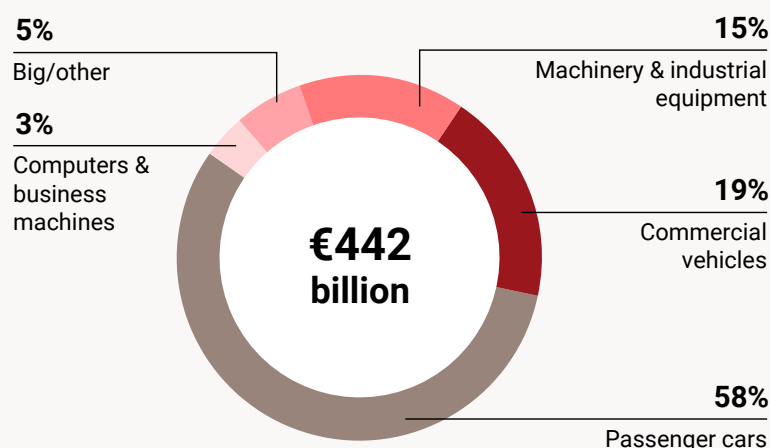
SAP ERP formed the financial backbone and was essential for governance. But contract management, billing processes, and asset lifecycle tracking varied across regions and business units. Some processes were highly manual and dependent on local expertise. Contract changes were handled inconsistently, and billing accuracy suffered when asset swaps or upgrades occurred. Finance teams invested significant effort into reconciliations at period close. Funding structures existed, but the linkage between portfolio composition and funding transparency was limited—making it harder to optimize cost of funds as the portfolio evolved.

The organization had reached a point where launching new offerings was possible, but scaling them globally with consistent control was not. That is the moment where many companies stall. Instead, this organization chose to build a scalable foundation.

“Most transformation risk lives in exceptions. Design the event model so exceptions shrink over time.”

The transformation was guided by a clear principle: build an integrated foundation across contract lifecycle, asset lifecycle, finance automation, and funding transparency—while keeping SAP as the financial backbone.

New leased equipment volumes per asset type in 2024



First, contract structures were standardized and modularized. Rather than allowing every region to reinvent contract logic, the organization established consistent patterns and controlled lifecycle events for common scenarios: onboarding, changes, renewals, and end-of-term options. This improved both speed and governance. Contract changes became trackable events instead of ad-hoc exceptions.

Second, asset lifecycle processes were aligned to create a consistent view of asset state and movement. The organization strengthened the relationship between physical events (deliveries, swaps, returns, refurbishments) and contract assignments. This reduced the “asset truth vs contract truth” gap that had been driving billing errors and disputes.

Third, finance processes were redesigned to be event-driven and explainable. Postings and billing effects were aligned with contract and asset events, increasing automation and reducing the need for manual adjustments. Importantly, the organization improved

traceability: finance teams could follow an audit trail from a posting to an invoice line, to a contract event, to an underlying asset event.

Finally, Asset based Funding transparency was strengthened. The organization improved its ability to link portfolio segments to funding structures and allocate funding costs in a clearer way. This enabled better profitability insight and supported strategic decisions as the portfolio mix shifted toward longer-term and more service-inclusive models.

The transformation did not merely deliver efficiency. It delivered confidence. The organization gained the ability to scale new offerings while maintaining financial and operational control.

“The breakthrough wasn’t a system replacement. It was connecting realities.”



Chapter 13:

Results

Why it matters: The real value of a leasing and XaaS transformation is not measured by isolated improvements, but by how the entire organization performs as a connected system. When contracts, assets, finance, and funding are aligned, complexity is no longer a constraint—it becomes a source of control and competitive advantage.

What you'll learn: How successful transformations translate into measurable impact across operations, finance, and strategy—and why true scale only emerges when all dimensions are integrated.

The impact of such transformations is best described across three dimensions: operational performance, financial governance, and strategic scalability.

Operationally, standardized contract patterns and controlled change events typically reduce cycle times and exception handling. The organization experienced improved consistency across regions, making it easier

to onboard customers, process changes, and manage end-of-term activities. Billing disputes reduced as asset assignments and contract state became better aligned. Asset redeployment readiness improved because lifecycle status became clearer and processes became more structured.

From a finance perspective, event-based postings and improved traceability reduced the manual reconciliation burden. Finance teams could close periods with greater confidence and spend more time on analysis instead of correction. Audit readiness improved because postings became explainable through consistent event chains.

Strategically, the organization gained a scalable foundation to expand XaaS and modern leasing offerings. Rather than treating each new offering as a one-off project, the organization could reuse standardized patterns and extend the portfolio more efficiently. Funding transparency supported growth planning by improving understanding of profitability after

cost of funds.

The broader lesson is that successful transformation is not about one feature. It is about connecting realities: the commercial reality of the contract, the physical reality of the asset, the financial reality of postings, and the capital reality of funding. When these realities align, scale becomes achievable.

“Transformation succeeds when operational reality, financial truth, and strategic intent finally operate as one system.”



Chapter 14: Common Pitfalls

Why it matters: Many leasing and XaaS transformations do not fail because of strategy, technology, or ambition—they fail because the operating model never truly changes. Organizations introduce new commercial models while keeping the underlying processes, asset logic, financial structures, and funding approach unchanged.

What you'll learn: The four most common failure patterns in leasing and XaaS transformations—and why overcoming them requires a fundamentally integrated approach across contracts, assets, finance, and funding.

Transformations in this space often fail for predictable reasons. One common failure mode is treating XaaS as a pricing change rather than an operating model change. Organizations launch a subscription offering but keep contract processes, asset processes, and finance processes unchanged. The result is a growing volume of exceptions and manual work.

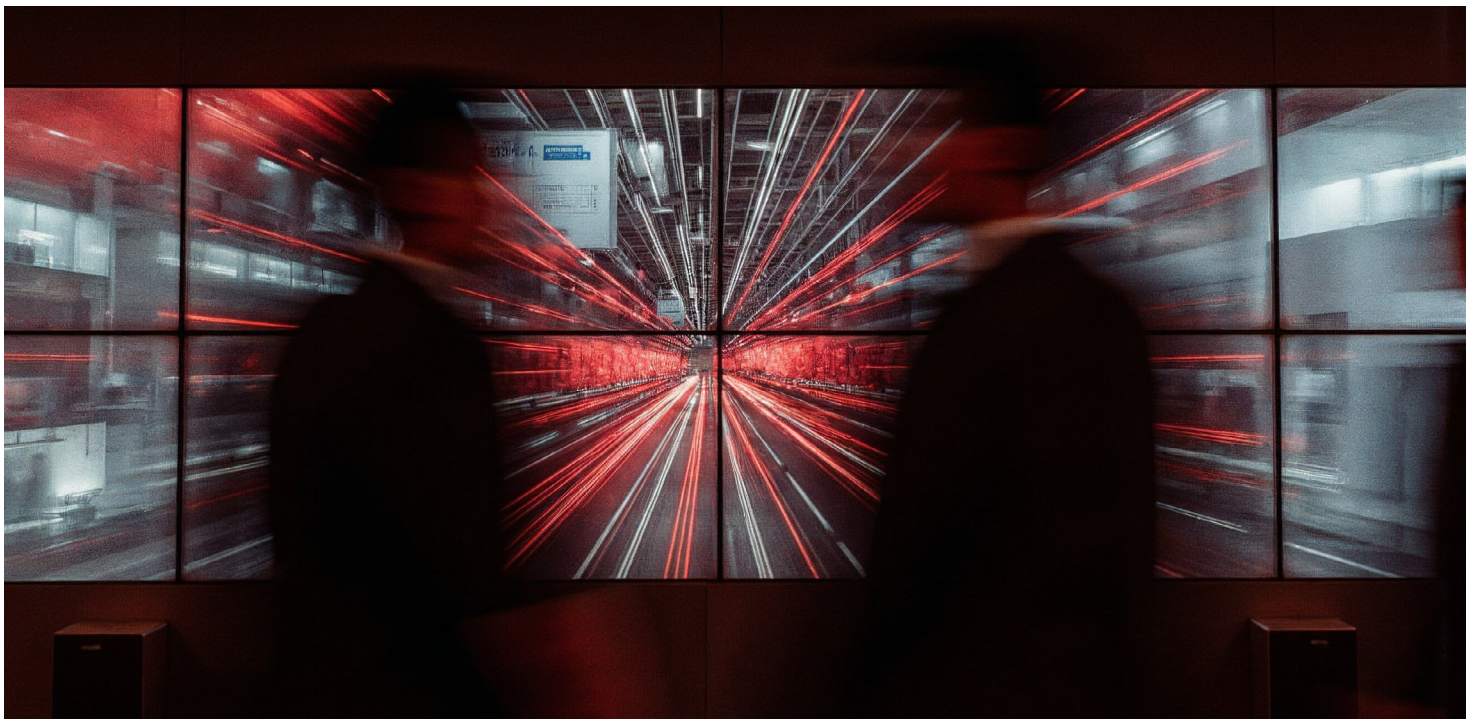
Another failure mode is modernizing customer-facing processes while leaving finance behind. Sales portals and contract tools improve the front-end, but the back-end still relies on reconciliations. Initially the business sees progress. Over time, finance becomes the bottleneck and risk increases.

A third failure mode is ignoring asset lifecycle complexity. XaaS success depends on lifecycle performance: utilization, downtime, maintenance, refurbishment, and redeployment. If asset lifecycle events are not integrated with contracts and finance, the organization cannot steer profitability effectively.

Finally, many organizations underestimate funding. As portfolios scale and asset retention increases, funding becomes a strategic constraint. Treating funding as a late-stage consideration can lead to expensive growth, limited refinancing flexibility, and weak profitability insight.

Avoiding these pitfalls requires a disciplined operating model and architecture approach. It is less glamorous than a new offering launch, but it is what enables sustainable scale.

“Most XaaS transformations don’t fail because of what companies change—they fail because of what they don’t.”



Chapter 15:

Roadmap — How to Start Without Disruption

Why it matters: The safest way to modernize is not a big-bang replacement, but a controlled rollout that proves value, hardens governance, and scales templates.

What you'll learn: A phased approach that aligns business outcomes with SAP-centric stability.

A practical roadmap is one that delivers value early and scales logically. In SAP-based organizations, a phased approach is often most successful.

The first phase is an assessment and target design. The goal is to establish a clear view of maturity across contracts, assets, finance, and funding. This includes identifying where the biggest friction points occur and where automation will deliver the highest returns. It also includes clarifying system roles and integration principles.

The second phase is a pilot focused on one segment: a product line, a region, or a contract model. This pilot implements standardized contract patterns, aligns key asset lifecycle events, and establishes event-driven finance integration into SAP. The pilot should be designed to generate measurable outcomes, such as improved billing accuracy or reduced manual posting effort.

The third phase scales the template across the portfolio. As the model expands, governance and master data discipline become increasingly important. The objective is to turn pilot success into a global operating standard.

This approach avoids the biggest transformation risk: the “big bang” program that attempts to change everything at once and fails under complexity. Instead, it creates a controlled path from learning to scale.

“Scale is not a phase. It’s a design requirement.”



How to start

If you are an OEM captive or leasing provider running SAP ERP and you are evaluating how to scale modern leasing and XaaS models with financial control, the most effective starting point is a structured assessment.

In a **30-minute Lease & XaaS Readiness Assessment**, we review your current setup across four dimensions: contract lifecycle, asset lifecycle, finance/subledger integration, and Asset based Funding transparency. The goal is not a generic discussion. The goal is to identify your top bottlenecks, clarify the fastest value levers, and define a practical next-step roadmap aligned with your SAP environment.

You do not need to prepare materials. The assessment is designed to be efficient and outcome-focused. You will leave the session with a maturity snapshot and a clear view of the most impactful actions — whether that is standardizing contract change events, improving asset-to-contract alignment, automating key postings, or strengthening funding transparency.

Book your 30-minute readiness assessment call.

BearingPoint — Lease&Rent | Assets&Funding
<https://bearingpoint.services/lease-and-rent/en/>

About BearingPoint

BearingPoint is an independent management and technology consultancy with European roots and a global reach. We help businesses transform by combining deep industry expertise with strong capabilities in strategy, operations, and technology. Dedicated SAP and Microsoft transformation units, a strong focus on AI, and outcome-based products enable us to provide tailored, innovative solutions that create measurable and sustainable value.

In addition to our core consulting operations, we run two joint ventures. Arcwide, our joint venture with IFS, specializes in business transformation enabled by IFS technology. BearingPoint North America, our joint venture with ABeam Consulting, focuses on consulting excellence and business transformation built on SAP.

BearingPoint works with many of the world's leading companies and public-sector organizations. Together with its strategic alliance partner ABeam Consulting, the firm brings together more than 15,000 professionals and serves clients in over 70 countries, delivering seamless business transformation, strengthening performance, and driving sustainable impact.

BearingPoint is recognized among TIME World's Best Companies and Forbes World's Best Employers. The firm is also a certified B Corporation, committed to responsible business and creating long-term value for organizations, people, and society.

For more information, please visit:

Homepage: www.bearingpoint.com

LinkedIn: www.linkedin.com/company/bearingpoint

Contacts

Yilmaz Tüfekci

Director

BearingPoint Products

yilmaz.tuefekci@bearingpoint.com

[LinkedIn](#)

Christoph Kühne

Business Development Manager

BearingPoint Products

christoph.kuehne@bearingpoint.com

[LinkedIn](#)