

WHITE PAPER · ENTERPRISE ARCHITECTURE

Design Principles For A Composable EA With A 2-Tier ERP Model

From the mindset of "one ERP for the whole enterprise" to a deliberately layered architecture: how to design the Digital Core, Tier 2 ERP, Integration, Data, and Governance layers for large enterprises and mid-sized company groups.

TIER 1

HEADQUARTERS (HQ)

Headquarters (HQ) & Core Business

INTEGRATION PLATFORM

TIER 2

Construction

Manufacturing

Logistics

F&B

Distribution



CONTENT OWNER

Lac Pham - CEO, S4 Consulting

Lead author. Research focused on Enterprise Architecture, ERP architecture, Business Capability and composable models.

Co-authored by: Nu Huynh & Sinh Pham - experienced consultants at S4 Consulting

Composable EA

Modular · API-led · Data-driven

2-Tier ERP

Core + Vertical ERP

Research Edition

Architecture · Governance · Economics

S4 Consulting

www.s4.com.vn

ERP@s4.com.vn

01 | CONTENTS

Table Of Contents

Architecting a Composable EA with a 2-Tier ERP Model

01	Contents	Page
01	Executive Summary Why 2-Tier ERP Can Be A Piece Of Composable EA	3
02	Architecture Thinking "Rolling Out ERP" Can Be An Architecture Mistake	4
03	Composable Enterprise Architecture How Is Composable EA Different From "Just Multiple Systems"?	5
04	Reference Architecture Designing A Composable EA With 2-Tier ERP	6
05	Integration & Data Integration, API, Events, And Data Architecture	7
06	Business Capability & Decision Framework When Should A Capability Sit In Tier 1 Or Tier 2?	8
07	Case 01 · Large Enterprise Enterprise Core And Vertical ERP	9
08	Case 02 · Mid-Sized Company Group Financial Backbone And Vertical ERP	10
09	License Economics & TCO Why The License Model Must Be Designed Together With Architecture	11
10	Governance & Operating Model Governance To Keep Composable From Becoming Many Silos	12
11	Implementation Roadmap Starting From Capability, Data Ownership, And Economics	13
12	Conclusion 2-Tier ERP Is An Architecture Pattern	14
13	Research Sources & FAQ Sources, research notes, and frequently asked questions	14

This white paper is structured around three questions: Why does ERP thinking need to change? How is a Composable EA with 2-Tier ERP actually designed? And in which contexts does this model create more value than rolling out a single ERP?

02 | EXECUTIVE SUMMARY

Why Can **2-Tier ERP** Be A Piece Of Composable EA?

Two-Tier ERP is not simply "using two ERPs." In a deliberately designed Composable Enterprise Architecture, the two ERP layers can play two different roles: a Digital Core focused on capabilities that require group-level standardization and control, and one or more specialized ERPs serving the speed, industry specifics, or operating model of individual business units.

The most important aspect of this model is not the number of ERPs, but the boundary of responsibility between the systems. SAP describes Two-Tier ERP as a strategy in which one system acts as a stable backbone while the second tier runs on separate, typically integrated, ERPs. SAP also points to common drivers such as M&A, business model change, and geographic expansion. Deloitte emphasizes that integration between the two tiers is essential to achieving real-time data, insight, and business agility. [\[SAP\]](#) [\[Deloitte\]](#)

The central thesis of this white paper: "Composable" does not mean more systems are always better. Composable means the enterprise can change one component without having to tear down the entire architecture.

PRINCIPLE 01**A Core With Purpose**

Keep financial, control, standard data, and strategic capabilities in a core with clear ownership.

PRINCIPLE 02**Tier 2 By Capability**

Move local, vertical, or deeply operational processes to a better-fit ERP when the benefit outweighs the integration cost.

PRINCIPLE 03**Integration Is Architecture**

APIs, events, data contracts, and master-data governance must be designed from day one - not bolted on after two ERPs are already running.

Two Practical Conclusions From The Research

For large enterprises: Tier 1 can be an ERP oriented toward Finance, Group Accounting, or a critical business domain with very high requirements for data scale, analytics, AI, and governance. Tier 2 can be industry-specific or business-unit ERPs, where the number of operational users can run into the thousands. In this model, license policy is no longer a software procurement detail - it becomes an architecture and TCO variable.

For mid-sized company groups: Tier 1 should typically serve as the group's financial backbone. Oracle NetSuite is a notable example thanks to OneWorld, which supports a multi-subsidiary structure, and Multi-Book Accounting. Tier 2 can use an industry-vertical ERP, where Acumatica is worth considering when the user count is large, since Acumatica's licensing model is not priced per user seat. For some business models, open-source ERP can also be considered if the enterprise has the right technology governance capability.

Methodology note: conclusions about the architecture model are the research's synthesis and recommendations. Product characteristics are drawn from official vendor documentation and should not be read as a claim that any single product always fits every enterprise.

03 | ARCHITECTURE THINKING

"Rolling Out ERP" Can Be An **Architecture Mistake**

Rollout is often seen as the safest way to expand an ERP: take a Tier 1 system that is running well at the group level and deploy the same model to a subsidiary. But "easy to roll out" technologically does not mean "optimal" from an architecture, business-fit, or TCO perspective.

A common reason is familiarity. IT, Finance, and operations teams already know the existing system, and templates and governance are already in place. This reduces upfront effort, but it isn't always a strong enough reason to push the same ERP down to every business unit.

"Easy to roll out" is a short-term benefit. "Fit and optimal across the whole lifecycle" is the criterion for Enterprise Architecture.

Mistake 1: Prioritizing Technology Familiarity

If a subsidiary's operating model is similar to the group's, rolling out Tier 1 can make sense. But when the business model, industry capability, or user population differs substantially, forcing that unit into the same template can mean the business has to adapt to the ERP rather than choosing the ERP based on capability.

Mistake 2: Ignoring Cost Optimization

ERPs like SAP S/4HANA or Oracle Fusion can be an excellent fit at the enterprise level, where finance, governance, analytics, and scale requirements are high. But deploying the full Tier 1 system to a subsidiary just because "it's already there" can create license, implementation, support, and change complexity far larger than actual need.

The issue is not choosing the cheapest software. What's needed is an assessment of **Business Capability Fit + Enterprise Control + Integration Complexity + 3-5 Year TCO**. A vertical ERP may be a better fit for a unit with industry-specific needs or thousands of operational users, if that benefit outweighs the integration and governance cost.

ROLLOUT-FIRST

"We already use it, so keep rolling it out"

Prioritizes familiarity and template reuse. Reduces upfront effort but can push enterprise complexity to where it isn't needed.

ARCHITECTURE-FIRST

"Which capability needs which system?"

Evaluates capability, industry fit, users, data, integration, and TCO before deciding to roll out or create a Tier 2.

Rollout Can Still Be The Right Choice

Two-Tier is not an argument against rollout. When a subsidiary's operating model is similar, requires the same governance, and standardization benefits outweigh the cost, rolling out Tier 1 remains a sound choice. The point is that rollout should not become the default.

04 | REFERENCE ARCHITECTURE

How Do You Design A **Composable EA** With 2-Tier ERP?

The reference architecture below treats ERP as part of the Digital Core and places Integration, Data, and Governance as independent layers. This design helps enterprises avoid turning Tier 1 into a "super-ERP" and Tier 2 into a system "locked in" to Tier 1.

TIER 1 DIGITAL CORE	Group Finance / Core Accounting / Strategic Business Domain. This is where the rules and data that must be standardized at the group level live: chart of accounts, legal entity, consolidation, financial controls, corporate master data, and strategic capabilities.
TIER 2 VERTICAL ERP	Industry / Business Unit ERP. Responsible for industry-specific operations, local operations, shop floor, project operations, distribution, retail, or any process that Tier 1 does not need to, or should not, own.
INTEGRATION & API/EVENT	Integration Platform. API gateway, iPaaS, event bus, orchestration, and data contracts. Do not let ERPs call directly into each other's databases without a governed architecture boundary.
DATA ARCHITECTURE	Master Data + Analytical Data. Defines the System of Record for customer, supplier, item, chart of accounts, legal entity, and transactions. The data platform serving BI/AI need not be the ERP database itself.
GOVERNANCE & SECURITY	IAM, SoD, Architecture Board, Release Management, Data Governance. This is the layer that keeps composability from turning into uncontrolled complexity.

Ownership Principles

Object	Tier 1	Tier 2	Principle
Legal Entity / COA	Owner	Consumer / mapped	Do not create a "shadow COA" without a genuine business reason.
Industry Master	Reference	Owner	A vertical ERP can own industry-specific masters, but a canonical mapping must exist.
Operational Transaction	Selective	Owner	Do not replicate every transaction if Tier 1 only needs a summary or the accounting impact.
Consolidated Finance	Owner	Contributor	Tier 1 should be the authoritative source for group reporting.

Recommendation: before selecting a product, draw a Business Capability Map and a Data Ownership Map. If these two maps are not yet clear, deciding "which ERP should be Tier 1" is still premature.

05 | INTEGRATION & DATA

Two ERPs Can Integrate Well If The Boundary Is Designed Right

Integration should not be seen as "a pipe connecting two ERPs." It is the mechanism that enforces the boundaries already defined by business capability and data architecture. Deloitte emphasizes integration as an essential element of two-tier architecture; Third Stage Consulting also notes that best-of-breed/composable opens up more choice but simultaneously increases the complexity of integration, solution architecture, and data management.

[\[Deloitte\]](#), [\[Third Stage Consulting\]](#)



Pattern	Use When	Strengths	Watch Points
Synchronous API	An immediate response is needed, e.g. checking a master record or credit.	Simple semantics.	Coupling, latency, availability dependency.
Event-driven	Notifying other systems of a transaction/state change.	Loose coupling, scalable.	Event ordering, idempotency, replay, monitoring.
Batch / File	Real-time isn't required, or with legacy systems / large data volumes.	Stable, easy to schedule and control.	Latency and reconciliation.
Data replication	Analytics, read models, or data products.	Reduces load on the transactional ERP.	Don't let the replica become a "shadow system of record."

Data Architecture: Where Do Most Failures Happen?

In multi-ERP projects, the issue isn't just "do the two systems have an API." The harder problem is how the two systems understand the same customer, item, legal entity, account, or transaction. If each ERP creates its own master data, an enterprise may end up with two customer codes for the same customer, two item codes for the same SKU, or two ways of calculating the same financial metric. In that case, integration can run correctly from a technical standpoint while being wrong from a governance standpoint.

Each domain should therefore have a single **System of Record** and a canonical model. Tier 1 need not own every master; Tier 2 should not become the "master of everything" just because the process happens to sit there. Data ownership must come with business accountability.

It should not be asserted that "the same ERP vendor's generation always integrates better." What can be said with more confidence is that shared accounting semantics, or shared compliance with clear integration contracts, may reduce some mapping and reconciliation effort. This remains a hypothesis to be tested in an architecture assessment.

06 | BUSINESS CAPABILITY & DECISION FRAMEWORK

When Should A Capability Sit In Tier 1 Or Tier 2?

ERP tiering should not be decided solely by the size of the subsidiary. A more sustainable approach is to tier by business capability, data criticality, and economics. A capability belongs in Tier 1 when it needs group-level standardization or forms the foundation of governance; it belongs in Tier 2 when it drives differentiation, has industry specifics, or requires a high rate of change at the business-unit level.

Criterion	Favors Tier 1	Favors Tier 2 / Specialist
Strategic control	Group-wide, high governance	Local / BU autonomy
Financial criticality	GL, consolidation, corporate finance	Operational sub-ledger / local process
Industry specificity	Common standardization	Highly industry-specific
Scale / data intensity	Big data, group analytics, AI-critical domain	Local high-volume operations
Change velocity	Low to moderate	High, needs fast release cycles
Localization	Global policy	Country / industry specific
User population	Corporate users / finance	Large operational workforce
Integration cost	Lower if native/shared	Only separate when business value outweighs integration cost

A Decision Scorecard Proposed For Vietnamese Enterprises

PLACE IN TIER 1 WHEN

The capability affects group reporting, treasury, consolidation, financial controls, enterprise master data, or a strategic business domain that requires high processing scale. Tier 1 must be able to function as a "stable backbone," not a repository for every custom requirement.

PLACE IN TIER 2 WHEN

The capability needs industry depth, local autonomy, fast deployment, or a large user population. Tier 2 must have APIs, data contracts, and the ability to evolve independently without breaking the group's control framework.

Checkpoints Before Deciding On 2-Tier

1. Capability

Who actually needs this capability? Is it a differentiator or a commodity?

2. Data

Who owns the data? Is real-time required, or is eventual consistency acceptable?

3. Economics

Do license + implementation + integration + support add up to a positive business case?

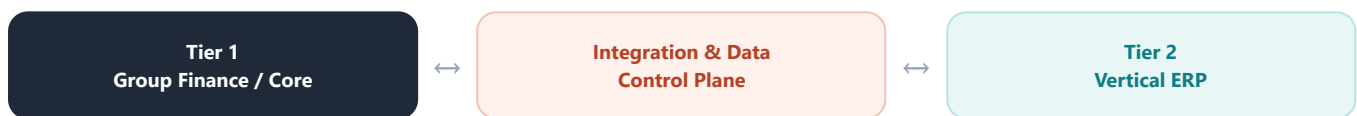
Panorama Consulting and Third Stage Consulting both warn of a downside to best-of-breed: multiple systems can deliver a better fit but bring greater integration, data-management, and governance complexity. [\[Panorama\]](#) [\[Third Stage Consulting\]](#)

07 | CASE 01 - LARGE ENTERPRISE

How Should Large Enterprises Design Tier 1 And Tier 2?

For large enterprises, Tier 1 does not necessarily need to be a "does-everything" ERP. A sound architecture might place Finance, Group Accounting, consolidation, and several strategic business domains into Tier 1, while Tier 2 focuses on vertical ERPs with deep industry capability, serving a very large operational workforce.

This is an expanded architectural interpretation of the Two-Tier ERP pattern already discussed by SAP, Deloitte, and EY. SAP describes Tier 1 as a backbone managing central functions, while Tier 2 can be designed for subsidiaries or specialized functions. EY gives the example of an acquiring company using two-tier ERP together with data orchestration to reduce integration cost and time during M&A. [\[SAP\]](#) [\[EY\]](#)



Tier 1: Where Is "Enterprise Depth" Needed?

Tier 1 fits capabilities the group does not want fragmented: group chart of accounts, consolidation, treasury, corporate reporting, master-data policy, intercompany governance, and domains with heavy requirements for transaction volume, advanced analytics, or AI. Some enterprises may place a critical business domain such as supply chain or manufacturing core into Tier 1 if that domain is itself the competitive advantage. "Finance is always Tier 1" should not be assumed by default; the more accurate principle is "a capability with a high level of enterprise criticality must have an authoritative core."

Tier 2: Vertical ERP For Thousands Of Users

At this layer, ERP can serve factories, stores, logistics networks, construction operations, field service, or business units with thousands of staff. As the user count grows sharply, user-based licensing can become an adoption barrier. This is why license architecture needs to be considered alongside solution architecture. A very capable ERP whose cost scales linearly with users can become a poor fit if every operational employee needs access.

ECONOMIC CONCLUSION

License Policy Is An Architecture Variable

At Tier 2, model the 3-5 year TCO: software license, user/access model, infrastructure, implementation, integration, support, upgrades, and the cost of governance. "Low license price" isn't enough - look at cost per operational participant and the ability to scale the user population.

Supporting evidence: Deloitte has described a two-tier cloud architecture with SAP S/4HANA Cloud as the core after consolidating 14 practices across 17 countries at Deloitte Africa. This is evidence that a two-tier architecture is feasible in complex enterprise environments, but not evidence that the same architecture suits every enterprise. [See the Deloitte case](#)



Oracle NetSuite: A Strong Candidate For Tier 1 ERP In A Mid-Sized Company Group

Potentially yes, if Tier 1 is defined primarily as the group's financial backbone and business units need deep operational ERPs. Oracle NetSuite is a fitting example for studying this pattern because OneWorld supports managing multiple legal entities/subsidiaries, and Multi-Book supports multiple sets of accounting records.

Oracle describes NetSuite OneWorld as organizing a hierarchy of legal entities and subsidiaries; Multi-Book Accounting allows multiple sets of accounting records to be maintained in real time from the same set of financial transactions. These are valuable capabilities when Tier 1 is designed as the group's financial backbone. [\[Oracle OneWorld\]](#) [\[Oracle Multi-Book\]](#)

TIER 1 · FINANCIAL BACKBONE

Oracle NetSuite

Well-suited for centralizing group finance, multi-subsidiary management, intercompany, consolidation, and reporting needs. This is an architecture recommendation, not a claim that NetSuite is always the best choice.

TIER 2 · VERTICAL OPERATIONS

Industry-Vertical ERP

Could be Acumatica or another ERP suited to manufacturing, distribution, construction, retail, F&B, or services. For enterprises with the right IT capability, open-source ERP can also be considered.

Why Does NetSuite Make Sense For A Group's Tier 1?

NetSuite OneWorld allows data to be organized by a hierarchy of subsidiaries; intercompany transactions can be processed across legal entities. Multi-Book allows multiple sets of books to be maintained for different accounting, regulatory, and reporting purposes. This is precisely the kind of "group backbone" capability a Composable EA typically wants to keep centralized. [\[Oracle\]](#)

Tier 2: Acumatica And The Unlimited Users Story

Acumatica publishes a pricing model that is not based on user seats, with unlimited users and cost primarily driven by applications, usage/resource requirements, and deployment. For a business unit with a large operational workforce, this is a variable well worth considering, since the ERP can be extended to many users rather than limiting adoption to avoid license cost. [\[Acumatica\]](#)

For a mid-sized company group, "Unlimited User" is not just a pricing advantage. It can change how the operating model is designed: an enterprise can bring more people directly into the ERP instead of adding another layer of spreadsheets, portals, or manual approvals purely to avoid buying more licenses.

Does Open Source Have A Place At Tier 2?

Possibly, for Tier 2, open source can fit when a vertical process needs deep customization, the enterprise has the right IT capability, and wants control over source code, data, and deployment. [\[Odoo\]](#)

09

| LICENSE ECONOMICS & TCO

Why Must The **License Model** Be Designed Together With Architecture?

In a Composable EA, software cost is no longer just "the ERP price." The enterprise is buying an architecture made up of multiple products, integration, a data platform, and operating capabilities. As a result, the license model can directly influence the decision to place a capability in Tier 1 or Tier 2.

SAP notes that two-tier can lower upfront procurement cost by using less expensive systems at the node level, but integration and support can raise ongoing cost. This is a critical point: 2-Tier should never be evaluated on license price alone. [\[SAP\]](#)

Cost Component	Single ERP	2-Tier Composable	Question To Ask
License	One vendor/platform	Two or more license models	Does cost scale by user, entity, transaction, module, or capacity?
Implementation	One large program	Two implementation streams	Can rollout be phased by capability?
Integration	Fewer endpoints	Higher	Who owns the integration platform and its SLA?
Data governance	Relatively centralized	More complex	Where is the System of Record?
Upgrade	One main roadmap	Multiple release cycles	Is version compatibility being governed?
User economics	Vendor-dependent	Can be optimized per tier	Does the vertical ERP fit a large workforce?

What Does Unlimited Users Mean For Architecture?

Acumatica publishes unlimited users and no per-user-seat charges under its current pricing model. This is especially notable when Tier 2 has hundreds or thousands of operational users. However, "unlimited users" does not mean "unlimited cost": enterprises still need to look at application scope, resource consumption, transaction volume, implementation, and support. The license advantage only holds value once it is built into a complete TCO model. [\[Acumatica\]](#)

S4's Recommended Principle

Don't choose an architecture because the license is cheap. Choose the architecture with the **lowest sustainable complexity**: cost low enough to scale, but governance strong enough not to create technical debt. For Tier 2, model "cost per operational participant" rather than just "cost per named user."

10 | GOVERNANCE & OPERATING MODEL

2-Tier ERP Fails When Governance Can't Keep Up

A multi-ERP architecture does not automatically become composable. If business units create their own master data, build their own point-to-point integrations, and decide their own processes without enterprise rules, the enterprise has simply traded "one monolith" for "many silos."

Third Stage Consulting emphasizes that best-of-breed can deliver a better fit but brings greater complexity in technical integration, solution architecture, and data management. Panorama also points out that a single ERP has the advantage of technical simplicity, while best-of-breed has the advantage of specialized functionality but requires integration. [\[Third Stage\]](#) [\[Panorama\]](#)

Governance Domain	Decision To Centralize	Proposed Owner
Architecture	Tier boundaries, integration standards, technology exceptions	Enterprise Architecture Board
Data	Master ownership, canonical model, lineage, quality	Data Governance Council
Security	IAM, SoD, privileged access, audit	CISO / Security
Release	Version compatibility, regression testing, release calendar	Platform / Application Owners
Finance	COA, intercompany, consolidation, statutory controls	Group Finance
Integration	API/event contracts, monitoring, retry, reconciliation	Integration Platform Team

Core Risk Register

Integration Sprawl

Point-to-point connections grow exponentially. Requires an integration platform and canonical contracts.

Duplicate Master Data

Multiple "sources of truth" for customer, item, or account undermine reporting reliability.

Process Fragmentation

End-to-end processes get cut apart; requires value-stream ownership and process orchestration.

Version Dependency

Two ERPs on different release tracks can break an integration contract; requires compatibility testing.

Shadow IT

Business units buy extra SaaS to fill Tier 2's gaps. Architecture review must be fast enough not to create a bypass.

Vendor Dependency

Composable does not eliminate vendor lock-in; the lock-in can simply move to the integration platform or data layer.

Operating Principle

Centralize what needs control; decentralize what needs speed. This is the principle best suited to a group with multiple business models. Tier 1 needs strong governance; Tier 2 needs bounded autonomy; Integration and Data need ownership that spans both tiers.

11

IMPLEMENTATION ROADMAP

Where Should Implementing A Composable EA With 2-Tier ERP **Begin?**

Don't start by choosing two ERPs. Start with capability, data ownership, and economics.

01

Map Business Capabilities

Identify which capabilities are enterprise-wide, which are vertical/local, and which genuinely create differentiation.

02

Define The Digital Core

Choose the capabilities that must have an authoritative core: group finance, consolidation, master governance, or a strategic domain.

03

Design Tier 2 Boundaries

Select vertical ERPs based on industry fit, local fit, user population, transaction volume, extensibility, and upgrade model.

04

Define The Data Contract

Establish the System of Record, canonical IDs, master synchronization, transaction ownership, and reconciliation rules before building integration.

05

Build The Integration Control Plane

Choose API/iPaaS/event/batch based on latency and coupling needs. Build in monitoring, retry, audit, and reconciliation from the start.

06

Model 3-5 Year TCO

Account for license, implementation, integration, data, support, upgrades, internal IT, and governance. Don't just compare license quotes.

07

Pilot One Value Stream

Select one business unit or value stream with clear boundaries to validate the architecture before scaling out.

"Composable" should be measured by the ability to make controlled change: adding a business unit, replacing a vertical application, or opening a new capability without having to redesign the entire Digital Core.

12 | CONCLUSION

2-Tier ERP Is Not A Destination. It Is An Architecture Pattern

The research concludes: enterprises should not ask "one ERP or two ERPs?" before defining the Digital Core and Business Capabilities. The right question is: which capabilities need centralized control, which need local/vertical autonomy, which data must be authoritative, and is the cost of fragmentation worth trading for agility?

In the context of Composable Enterprise Architecture, Two-Tier ERP becomes a pattern worth considering when an enterprise needs both a stable core and specialized ERPs that can change faster. SAP, Deloitte, EY, Panorama, and Third Stage Consulting each offer different perspectives, but they converge on one point: multiple systems can create business fit and agility, but integration, data, and governance are the price that must be managed. [\[SAP\]](#) [\[Deloitte\]](#) [\[EY\]](#) [\[Panorama\]](#) [\[Third Stage\]](#)

CASE 01 · LARGE ENTERPRISE

Tier 1 = Enterprise Core

Finance / Group Accounting, or a strategic domain with high requirements for scale, data, analytics, and AI. Tier 2 = vertical ERPs for business units and a large operational workforce.

License economics is an architecture factor.

CASE 02 · MID-SIZED GROUP

Tier 1 = Financial Backbone

NetSuite OneWorld is a strong example for multi-subsidiary and Multi-Book. Tier 2 = vertical ERPs such as Acumatica, or open source where appropriate. **Unlimited-user economics can determine the ability to scale.**

Research Conclusion

This research argues that the future of ERP architecture is not necessarily "a more powerful ERP," but an Enterprise Architecture that knows how to put the right system in the right role. For large enterprises, what matters is keeping a core deep enough for financial governance or a strategic business domain, while letting vertical ERPs serve thousands of users without license cost becoming a barrier. For mid-sized company groups, a financial backbone such as Oracle NetSuite can be paired with vertical ERPs to create an architecture that balances standardization and flexibility.

From an architecture standpoint, integration and cost model are two parts of the same problem. An ERP with great features but a poor fit for user economics, or a "composable" architecture that lacks data governance, can both create long-term costs larger than their initial benefit.

Scope of this document: this white paper focuses on the architecture pattern and decision framework. Product characteristics are cited from official sources. Recommendations to use NetSuite as Tier 1 and Acumatica/open source as Tier 2 are architecture recommendations that should be validated for each enterprise, and are not official statements by Oracle, Acumatica, or Odoo.

12

RESEARCH SOURCES & FAQ

What Sources Were Used To Build This White Paper?

Tier A - Primary / Vendor: SAP - Two-Tier ERP definition, deployment options, and SAP Learning; Oracle - NetSuite OneWorld and Multi-Book documentation; Acumatica - pricing/unlimited users; Odoo - open-source model.

[SAP Two-Tier ERP](#) · [SAP Deployment Options](#) · [Oracle OneWorld](#) · [Oracle Multi-Book](#) · [Acumatica Pricing](#) · [Odoo](#)

Tier B - Big Four: Deloitte - two-tier cloud architecture and API-led Composable Enterprise; EY - rapid M&A IT integration with two-tier ERP; KPMG - Composable Enterprise Architecture and the intelligent organization.

[Deloitte - Cloud as a Competitive Edge](#) · [Deloitte Digital - Composable Enterprise](#) · [EY - Rapid M&A IT Integration](#) · [KPMG - Future of Corporate Services](#)

Tier B - Independent ERP Advisory: Eric Kimberling / Third Stage Consulting - best-of-breed, composable ERP, integration/data complexity, and technology-agnostic ERP strategy; Panorama Consulting - best-of-breed, two-tier ERP, and the trade-offs between single-system simplicity and specialized functionality.

[Third Stage - Best-of-Breed / Composable](#) · [Third Stage - Strategy Decisions](#) · [Panorama - Best of Breed](#) · [Panorama - Two-Tier Trends](#)

Additional research: Gartner - "Two-Tier ERP: A Flexible Consolidation Path for Complex Organizations," published May 2026; IEEE Access research on Composable Enterprise Architecture. These sources are used as contextual support rather than as sole evidence for product-specific recommendations.

[Gartner - Two-Tier ERP](#) · [IEEE Access](#) · [Composable Enterprise Architecture](#)

Lead Author: Lac Pham - CEO, S4 Consulting.

Contributors: Nu Huynh - Director, S4 Consulting & Sinh Pham - Consultant, S4 Consulting

Role: Lead author, responsible for the structure, argument, and synthesis of this white paper.

Scope: This document is a research and architecture discussion piece. S4 Consulting is its editor, not a source confirming the capabilities of the vendors mentioned.

FAQ

Does Tier 1 have to be the largest ERP?

No. Tier 1 should be the system that plays the enterprise-backbone role for capabilities that need centralized control. In some enterprises that's Finance; in others, it could be a strategic business domain.

Why does license matter so much for Tier 2?

Because Tier 2 typically serves many operational users. If license cost scales sharply with users, an enterprise may limit adoption or add another intermediary layer of systems. An unlimited-user model can change the economics, but the full TCO still needs to be accounted for.

Is NetSuite suitable as Tier 1 for a mid-sized company group?

It can be, when the goal for Tier 1 is a financial backbone. OneWorld supports multi-subsidiary structures and Multi-Book supports multiple sets of accounting records. The final choice still depends on capability, country/localization, TCO, and architecture.

Can Acumatica serve as a vertical Tier 2 ERP?

It can be a candidate worth evaluating, especially when a business unit has industry-specific processes and a large user population. Acumatica publishes unlimited users, with pricing based on applications, usage/resources, and deployment. Actual fit must be validated by industry and transaction profile.

This white paper was authored by the S4 Consulting advisory team. It was built from public-source research and ERP consulting experience; it is not an official document of SAP, Oracle, Acumatica, Odoo, Deloitte, EY, KPMG, Third Stage Consulting, or Panorama Consulting.

Designing ERP Is Not Just About Choosing Software. It's About Designing How The Enterprise Can Change.

This document presents a reference model for designing Enterprise Architecture along composable lines, in which Two-Tier ERP is treated as an architecture pattern to use when an enterprise needs to balance enterprise control, vertical specialization, data consistency, and economic scalability.

ENTERPRISE ARCHITECTURE

Business Capability · Digital Core · 2-Tier ERP

ARCHITECTURE

Integration · Data · Governance

ECONOMICS

License · TCO · Scalability

A PRACTICAL CASE IN VIETNAM

Implementing A 2-Tier ERP Model With Acumatica

Acumatica can be considered as a Tier 2 ERP for business units that need deep operational capability, the ability to scale users, and economics that fit the group's operating model.

[Learn about this implementation model](#)