How to Design Enterprise Applications That Are Composable by Default

Published 26 April 2021 - ID G00738874 - 15 min read

By Analysts Yefim Natis, Massimo Pezzini, Anne Thomas

Initiatives: Software Engineering Strategies; Applications and Software Engineering Leaders; Software Engineering Technologies

Composable enterprise applications deliver business adaptability, but design of composability is not well-understood. Applications and software engineering leaders aiming for business agility should use the Gartner Composable Business Index to assess and advance the agility of their applications.

Overview

Key Findings

- Composable applications are the key to achieving agility, but organizations lack methods of measuring application composability.

- Organizations face unknown risks because of the gaps in their process and skill preparedness for composable software engineering.

- Without proven best design practices for composability, applications and software engineering leaders struggle to deliver consistently composable software.

Recommendations

For applications and software engineering leaders aiming to help accelerate business agility:

- Assess the composability of applications by using the Gartner Composable Business Index as a guide.

- Evaluate your organizational preparedness for the composable enterprise before settling on a future roadmap.

- Design new applications as composable by default by optimizing for modularity, autonomy, orchestration and discovery.

Strategic Planning Assumption
By 2024, 70% of large and medium organizations will include composability in their approval criteria for new application plans.

**Introduction**

Business resilience is essential in the face of disruptions like the global pandemic, changing regulations or geopolitical realignments. Resilience of a business can be improved (or hindered) by software. Software agility therefore becomes a central challenge for applications and software engineering leaders. But they soon discover that neither business nor IT can address this challenge alone. A failure to work together in multidisciplinary fusion teams leaves organizations lagging behind their more agile competitors. Application composability principles help facilitate success of fusion teams. But the best composability practices are not well-established.

How can applications and software engineering leaders create a dependable plan of action that aligns with the demands on the business for a fast pace of change? Gartner's Composable Business Index defines the fundamental characteristics of composable enterprise experience and is designed to guide organizations to the best composable engineering practices (see Figure 1).
Apply the Gartner Composable Business Index to Assess Composability of Existing Applications and to Design Applications With Composability in Mind

**Modularity**
- Is the method of modularity clearly defined?
- Do components have singular, clear and complete business identity?
- Is the component data fully encapsulated?

**Discovery**
- Does component documentation and metadata support granular discovery and introspection?
- Is it machine and human readable?
- Is it accessible at run-time?

**Autonomy**
- Is the impact of change well-isolated?
- Is the component data integrity well-internalized?
- Are components replaceable?

**Orchestration**
- Do component interfaces support all prevailing standards, protocols and architectures?
- Do they support access for tracking, governance and regulation?

Source: Gartner 738874_C

### Analysis

**Assess the Composability of Applications**

The Gartner Composable Business Index (the Index is defined in Toolkit: Composable Business Index From the 2020 Gartner IT Symposium/Xpo Keynote) applies to many different facets and contexts of modern business, including its technology, organization, ecosystem, marketing strategies and business models. In each area of business the objective of increased agility is the same, but its core tenets apply differently. For example, ask yourself: What are the building blocks that support composability for organization, ecosystem, marketing strategy and business model?

Here, we apply and specialize the Index to guide applications and software engineering leaders to success in assessing or delivering well-functioning composable enterprise applications.
The essential value of composability is the readiness to be recomposed: The readiness for fast, safe and efficient change to match the demands for efficiency, resilience and adaptability of business. Although “fast” is often an enemy of “safe” and “safe” can be a challenge to “efficient,” the design principles discussed in this research can advance all three in a balanced manner.

The principles of the composability Index would be familiar to an application architect. Decoupling, microservices, events, orchestration, choreography, modularity, backward compatibility, encapsulation and decomposition are all concepts that have been part of software architecture practices for quite a while; separation of concerns, bounded context and cohesion are parts of domain-driven design (see Note 1). The composable enterprise model essentially reapplies the core and advanced principles of service-oriented architecture (SOA) and mesh app and service architecture (MASA) to the environment with a key new constraint that the “services” must represent a well-defined, self-contained business value.

The defining advance of the composable enterprise architecture, compared to the previous modular software architectures, is its new and essential focus on business-IT fusion teams in place of the traditional teams of toward delivering business technology innovation. All components of the composable application architecture therefore must be accessible to both business and technology-oriented “composers.”

The four core tenets of the Gartner Composable Business Index, applied to business software, drive the consistency of the composable enterprise application experience:

- **Modularity** is central to composability, whether it is applied to software, organization or business model planning. Whatever building blocks are provided for the composition define the character of the resulting architecture. One of the key differentiators of the composable enterprise experience is that application design and redesign (composition and recomposition) are performed with direct participation of business and technology professionals operating as multidisciplinary “fusion” teams (see Fusion Teams: A New Model for Digital Delivery). To best meet this essential requirement, a composable enterprise application must be a cohesive set of well-defined packaged business capabilities (PBCs), the independent composition “building blocks” (see Innovation Insight for...
Composable Modularity of Packaged Business Capabilities. Well-implemented modularity will produce the PBCs that are small enough to maximize agility, but large enough to contain integrity.

- **Discovery** readiness is about making it easy for teams (especially fusion teams in this context) to find the components (here, the PBCs or their collections), determine if it will be useful for them, and having enough documentation for the teams to be able to integrate easily. Readiness for discovery implies the clarity, completeness and usability of documentation and other metadata. Advanced discoverability includes support of introspection to establish operational characteristics of the component, such as the metrics of performance and resource consumption. At runtime, some discovery metadata may be used for process control.

- **Autonomy** is the characteristic of the modularity that maximizes self-contained independence of the components (here, the PBCs). In the business application software design, the autonomy implies the ability of the components to fully manage their internal integrity, impose minimal burdens on other components when they are changed or replaced, suffer minimal dependency on changes in other components, and be capable of operating alone or in new, unanticipated combinations. Complete autonomy is rarely achievable, but each improvement of this measure will translate to the overall improvement of the organization's composability experience.

- **Orchestration** refers to the preparedness of the components (PBCs) for composition, integration and governance in development and runtime environments. In composable business application software, “orchestration” reflects the openness of the components to interact with other software, and the support of the prevailing models and protocols of communication. Orchestration also measures how well the components are equipped for monitoring, tracking, securing and the DevOps operations and other forms of governance. In other words, it is the gauge that measures the quality, openness, safety and controllability of the encapsulation.

Applications and software engineering leaders should use the Index as the foundation for consistent rollout of their strategy toward the agility of the composable enterprise, including process practices, technologies and architecture. In particular, they should use the Index in early technology decisions and later in strategic planning. By evaluating application software and tools on the four dimensions of the Index, applications and software engineering leaders can:

- **Examine software engineering tools** for their ability to create well-composable components and applications.

- **Assess deployed custom applications** to determine their inherent composability and associated potential contribution or resistance to the organization's composable software agility.

- **Help application leaders in their appraisal of vendor applications** to develop a deeper understanding of their potential for agility and innovation before making a purchase decision.

Based on these assessments, applications and software engineering leaders can then:
Evaluate Your Preparedness for the Composable Enterprise

Composition of applications is a benefit to any business that needs to change quickly, safely and more efficiently. But the urgency and preparedness for composability differ between organizations. Applications and software engineering leaders must calibrate their plans to reflect the specific context in their organization.

Moving to modular composable architecture too fast or too soon (without an assessment of available skills and resources, clarity of purpose and a roadmap) will lead to a chaotic environment and project failures instead of the expected optimization.

To establish a realistic and sufficient initial plan of adopting composability, applications and software engineering leaders should:

- Understand the anticipated business demands for change
- Consider the urgency and costs of pursuing greater application agility
- Take into account the available tools and skills
- Assess the organization's overall preparedness for composability

The following are some signs of preparedness for composable enterprise application engineering:

- Product-centric (versus project-centric) technology delivery culture. ¹
- Maturity in agile software development practices and engineering. ²
- Experience publishing and supporting externally published APIs. ³
- Experience utilizing low-code development and integration tools. ⁴
- Experience forming multidisciplinary (fusion) teams. ⁵
The sum of these considerations will determine the degree of composability you should plan to attain in your initial roadmap. It will also indicate how much external help might be required to assure success of your composable application initiative.

**Design New Applications as Composable by Default**

The key differentiator of composable enterprise applications from other forms of modularity (such as SOA and MASA) is its audience. Who is doing the composing defines what the building blocks and composition tools must be. Most prior design models and engineering tools targeted either IT professionals or business users (“citizen developers”). However, the pace of business change that calls for composability is such that neither can manage it alone. To embrace the composable enterprise model, applications and software engineering leaders must enable multidisciplinary fusion teams to be the designated “composers.” (This shift of software engineering focus — from IT professionals on their own to the multidisciplinary business-IT fusion teams — is a key driver of the democratized composable enterprise.) The building blocks (PBCs) and the composition tools are designed for both the business and IT professionals, working together, drawing on their complementary talents.

Neither business nor IT professionals can meet the demands of the increasing pace of business change alone. Composable enterprise architecture must serve the multidisciplinary (fusion) teams of business and IT professionals as its software engineering constituency.
Business-Centric Modularity of Composable Enterprise Applications

The modularity of PBCs encapsulates business capabilities to support application design and development by fusion teams. The design of PBC modularity inherits some characteristics from both the monolithic business applications and the more technical microservices architecture (see Figure 2):

- The project-delivered monolithic applications are relatively easy to manage, since all their parts are tightly connected and always change together. These applications are managed as one unit. But for that exact reason they are hard to change.

- The product-delivered microservices-based applications are easier to change (by reconnecting the microservices and user apps in a new way), but harder to manage (since each microservice may be changing independently).

- The PBC-based applications are assembled of components that encapsulate notable business objects (bank account, purchase order) or processes (loan approval, employee onboarding). They bundle all APIs and event streams required to manage the full life cycle of their managed entity. PBCs thus behave as if they were mini monolithic applications. The full applications now are assembled of the PBCs and have the agility comparable to the microservices-based designs. Thus PBC-based applications have manageability that is similar to monolithic applications and agility that’s similar to microservices-based designs.
As with all innovation and change, applications and software engineering leaders must proceed incrementally, understanding the balance of their skills and resources versus the demands of a new approach; making smaller, less disruptive changes first. Not everything needs to be composable (especially at the beginning). All PBCs don’t have to be of the same size or internal design. Applications and software engineering leaders should consider the recommendations below as a long-term roadmap plan, noting that the initial adherence to just a subset of the recommendations will already produce a measure of new adaptability and create a foundation for further advance.

Applications and software engineering leaders achieve consistently effective composability of applications by optimizing them in accordance with the four principles as defined in the Gartner Composable Business Index: modularity, discovery, autonomy and orchestration.

To optimize modularity:

- **Partition** application functionality into components with clear and complete business responsibilities (PBCs) by encapsulating:
  - Business objects like account or product
  - Physical objects like a turbine or an engine
  - State of a process like order to cash or a loan approval
  - A well-defined data/analytical service like a credit check or a price forecast

- **Limit** the individual PBCs to a single isolated business responsibility that is big enough to have a clear business purpose, yet small enough to eliminate cross-functional entangling within a PBC.
  - Early designs, especially when modernizing existing applications, may produce larger PBCs, to be refined over time.

- **Implement** an exhaustive set of interfaces (APIs and event streams) per PBC to encapsulate the full life cycle of its subject business object or process. A well-encapsulated (packaged) business capability eliminates any need for external access to its internal data.

To optimize discovery (see Figure 3):

- **Establish or join** a catalog/marketplace that provides some or all of registration, search, governance, certification, security, versioning, tracking and other governance functionality for both development
and runtime.

- Backstage, SwaggerHub and RapidAPI are current examples of supporting technology (some are open source; see Create API Portals That Drive API Adoption Among Internal and External Developer Communities).

- **Provide** clear and consistent documentation for the PBCs and their component APIs, including a comprehensive collection of machine-readable metadata for automated and AI-based interpretation at development and runtime.

- **Catalog** outer APIs of “legacy” applications alongside the PBCs to support participation of older applications in composition projects. Also, include user interface (UI) components, data schemas and other types of assets useful in design of PBCs or composition of applications.

- **Register** named collections of PBCs to optimize search and provisioning. Some collections may represent functionality set for:
  - A particular job role (hiring manager)
  - Job responsibility (candidate interviewing)
  - An end-to-end process life cycle (new hire onboarding)
  - An entire business unit (HR)

- **Coordinate** the choice of the catalog/marketplace and the application composition technologies (see Innovation Insight for Application Composition Technology).

- Compatible or shared specifications between the composition technology and the PBC discovery catalog will optimize operation of the entire composable environment, including efficiency, consistency, security, integrity and accountability.
Catalog/Marketplace Curates the Discovery and Management of Composition Assets

To optimize autonomy:

- **Implement** complete encapsulation of the internal data and process of the PBC to ensure protection of its internal integrity.
  - Restrict direct access to internal resources of the PBC.

- **Reduce**, to a minimum, external dependencies in either direction.
  - Aim to make the remaining dependencies optional.

- **Limit** all outer interactions of a PBC, in both directions, to only intermediated, mediated methods.
  - Choose event-driven interactions, where the business logic supports it for its native intermediation via event brokers.
To optimize orchestration (composition):

- **Strictly forbid** access, in either direction and for any purpose, via undocumented interfaces.
- **Maximize** backward compatibility when versioning a PBC, to eliminate all avoidable impact on related modules and processes.
- **Ensure** the PBC remains UI-agnostic, even when a UI is provided in its packaging.

Evidence

1. Strategic Roadmap for Becoming a Digital Product Delivery Organization
2. The Questions Enterprise Architects Ask Most Often About Agile and Lean Methods
3. Critical Capabilities for Full Life Cycle API Management
4. Critical Capabilities for Enterprise Low-Code Application Platforms
5. Fusion Teams by the Numbers: An Empirical Analysis of Digital Business Teams

Additional research by: Gary Olliffe
Note 1: What Is Domain-Driven Design?

Domain-driven design (DDD) is a software design philosophy for building complex systems in which the design is centered on the real-world business domain that the software serves. It strives to tame system complexity to ensure that the software can be easily changed or extended as the business domain evolves. Learn more at:

- Domain-Driven Design Quickly, InfoQ.
- E. Evans, Domain-Driven Design Reference, Domain Language.

Recommended by the Authors

- Strategic Architecture Roadmap for Composable Enterprise Applications (Presentation)
- Use Gartner’s Reference Model to Deliver Intelligent Composable Business Applications
- Kick-Start Your Composable Business Journey With 2 Key Strategies
- Toolkit: Composable Business Index From the 2020 Gartner IT Symposium/Xpo Keynote
- Innovation Insight for Application Composition Technology
- 2021 Strategic Roadmap For The Composable Future Of Applications
- Future of Applications: Delivering the Composable Enterprise
- Innovation Insight for Composable Modularity Of Packaged Business Capabilities
- Application Leaders: Master Composable Enterprise Thinking for Your Post-COVID-19 Reset
- The Applications of the Future Will Be Founded on Democratized, Self-Service Integration