Quick Answer: How Can Existing Applications Participate In Composable Architecture?

Published 9 August 2021 - ID G00752704 - 6 min read

Yefim Natis

Initiatives: Applications and Software Engineering Leaders

Composability is an emerging trend in architecture of new and future business applications. Software engineering leaders can expand the scope of their composable experience, or prepare for it, by leveraging the APIs, data and features of existing applications.

Quick Answer

How can existing applications participate in composable architecture?

- **Deploy platform technology to facilitate composition.** Establish an API catalog/marketplace — the critical centerpiece of the composable application environment. Deploy low-code technology that has access to the catalog content, supports the typical API formats, such as REST or event streams, and includes tools for UI development. Support fusion teams as “composers” of new application experiences, utilizing low-code composition technologies.
- **Develop business APIs to include existing applications in new compositions.** Identify existing application(s) that have features or data that are wanted for composition initiatives. Start small. Grow over time. Catalog the existing business-centric APIs of the selected applications, where they exist. Use integration or API management technology to design and create business-centric APIs in front of the applications that do not offer them. Refine the APIs over time, relying on API management technology to track and propagate API changes to composed applications.
- **Build up component metadata to increase the intelligence of composition management.** Provide an abundance of suitable identifying, introspective and operational metadata for the catalogued business APIs to support effective discovery and composition. Maintain and expand the metadata over time.

- **Use composability design principles to deliver more effective compositions.** Use Gartner's Composable Business Index to assess and improve the composable quality of the catalogued business APIs. Group related APIs into API products and packaged business capabilities (PBCs) to increase autonomy of the set, by following the principle "what changes together, stays together."

More Detail

Deploy platform technology to facilitate composition.

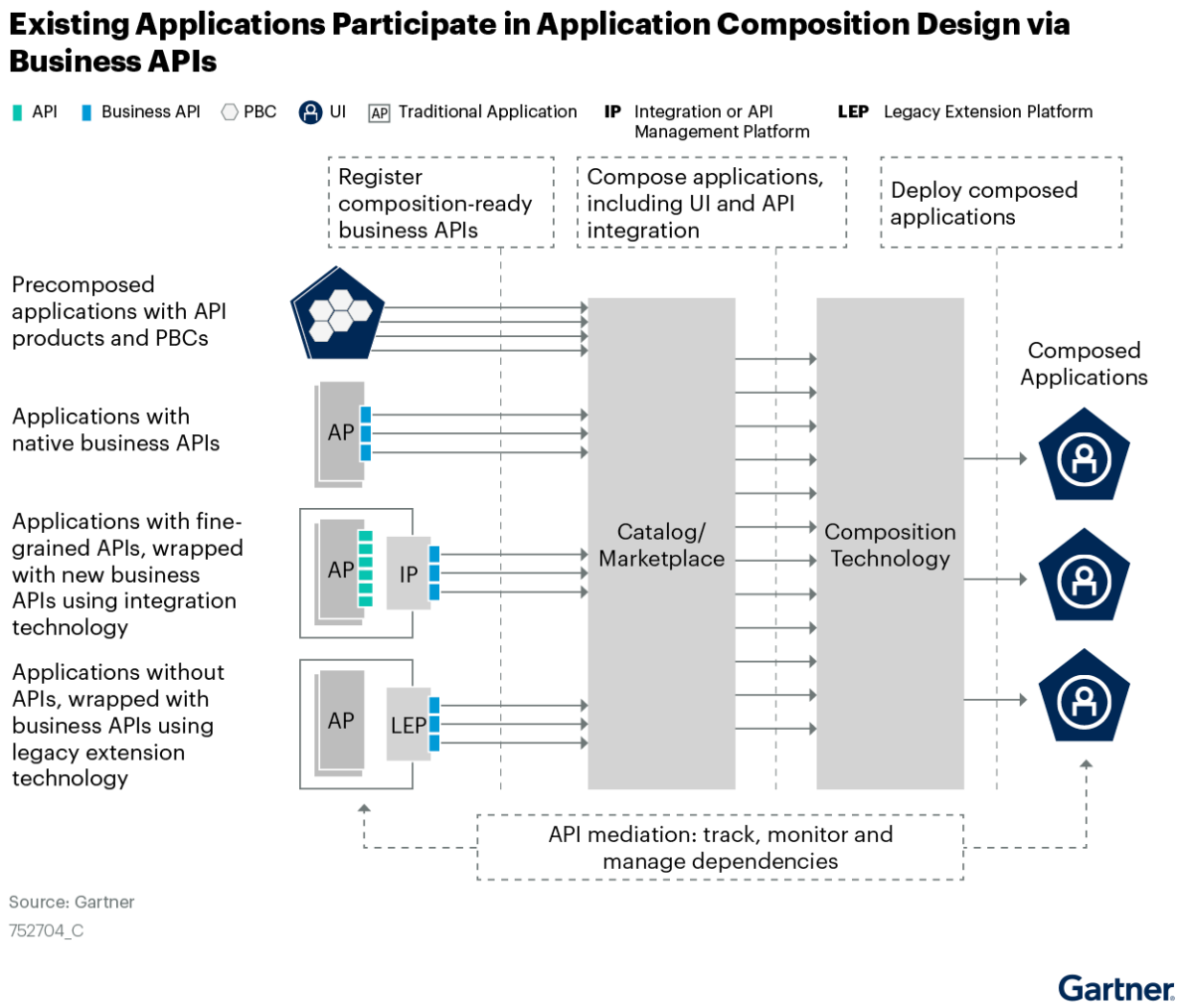
The composition process relies on consistent and managed access to a growing collection of registered, certified and protected business-centric APIs of various types. This makes the catalog/marketplace the centerpiece of the composable application architecture. [1] Early API registrations may center on just the security, access and discovery of the APIs, but more advanced marketplaces will support certification, vetting, optimization, tracking and automation as well. Today's API management platforms [2] and marketplace specialist offerings like RapidAPI, Pronovix and Achieve Internet, can be used to begin building a custom catalog/marketplace. Some software engineering teams develop their own, not finding sufficient functionality in available tools. This is a complex undertaking and should not be taken lightly. For some existing applications, older API standards, like SOAP, may have to be supported along with the more modern alternatives, like OpenAPI, GraphQL and AsyncAPI. Depending on the nature of the applications, registration of event interfaces may be essential or optional.

Composition technology must have the ability to access the APIs in the selected catalog/marketplace and build processes and user interfaces that utilize those APIs. Because composable application architecture targets fusion teams as primary users, we recommend using low-code technology to ensure accessibility to many personas. Some integration platforms (iPaaS [3]) and the more advanced LCAPs [4] and MXDPs [5] can deliver most of this essential set of capabilities for composition, but the catalog/marketplace will likely have to be created separately. Maximize integration of the selected composition tools and the catalog/marketplace, so the catalog's metadata can be fully interpreted by the composers.

Develop business APIs to include existing applications in new compositions.

Applications that are composable by their initial design are emerging in the form of API-centric SaaS and API products. [6] But most applications deployed in mainstream organizations are not so designed. Some existing applications are in demand for their data and services and need to be adopted for inclusion in composable architecture (see Figure 1).

Figure 1: Existing Applications Participate in Application Composition Design Via Business APIs



Applications or software engineering leaders responsible for composition may need to adopt both custom and vendor applications, including SaaS, to participate in application composition:

- Some applications will have well-defined business-centric APIs, ready for registration. [6]

- Many will have APIs that are too technical and are not suitable for business-centric composition, as the targeted "composers" are multidisciplinary fusion teams that include business designers along with technology professionals. [7] Those will require that you use integration technology to aggregate and orchestrate the functionality and data of the application to create new, more-coarse-grained, business-centric APIs. [8]

- The oldest of the applications may not have any APIs at all. Look for tools, like OpenLegacy or Rocket Software, to create the wrapper APIs for those applications, [9] though a better approach in the long term is to modernize and replace them (the lack of APIs is likely an indication of an outdated internal architecture as well).

The full effect of composable architecture is derived from the well-defined API-encapsulated (packaged) business capabilities. Wrappers of existing (potentially monolithic) applications alone will rarely reach that quality of service, but they are essential for incorporating existing functionality in future application compositions.

Build up component metadata to increase the intelligence of composition management.

The more information about the registered APIs that is provided with the registration, the more successful the process of discovery and management of the APIs can be. Give priority to collecting all suitable metadata that documents the interfaces, performance, operations and business content of the APIs. Train the potential "composers" in the operational implications indicated in the API metadata [10] (for example, an API with a 5x8 support model should not be used in a composed application that needs 8x7 support). To support automation, collect machine-readable metadata. [11]

Use composability design principles to deliver more effective compositions.

Use the best practices of API-first integration [12] and the core design principles of composable architecture [13] to improve the architectural fit of the business APIs of existing applications (by adjustment or redesign of the APIs, and modernization or replacement of the older applications):

- Modularity for cohesive and predictable representation of individual business capabilities of applications

- Autonomy for protection of integrity and safety of change of the encapsulated/packaged business capabilities

- Orchestration for effective managed compositions and recompositions of business capabilities to form application experiences
- Discovery for secure governance of the portfolio of the composable packaged business capabilities, reducing redundancies, fragmentation and other inefficiencies

Additional research provided by Gary Olliffe.

Recommended by the Authors

How to Design Enterprise Applications That Are Composable by Default

Kick-Start Your Composable Business Journey With 2 Key Strategies

Create API Portals That Drive API Adoption Among Internal and External Developer Communities

Use Gartner's Reference Model to Deliver Intelligent Composable Business Applications

Strategic Architecture Roadmap for Composable Enterprise Applications (Presentation)

How to Successfully Implement API-First Integration

How to Derive Value From APIs Using API Marketplaces

Evidence

[1] How to Design Enterprise Applications That Are Composable by Default

[2] Magic Quadrant for Full Life Cycle API Management

[3] Magic Quadrant for Enterprise Integration Platform as a Service

[4] Magic Quadrant for Enterprise Low-Code Application Platforms

[5] Magic Quadrant for Multiexperience Development Platforms

[6] Kick-Start Your Composable Business Journey With 2 Key Strategies

[7] Fusion Teams: A New Model for Digital Delivery

[8] The Applications of the Future Will Be Founded on Democratized, Self-Service Integration

[9] Use Continuous Modernization to Build Digital Platforms From Legacy Applications

[10] Quick Answer: What Should Be in an API Usage Agreement?

[11] Innovation Insight for Self-Integrating Applications

[12] How to Successfully Implement API-First Integration

[13] Toolkit: Composable Business Index From the 2020 Gartner IT Symposium/Xpo Keynote