ca technologies
A Broadcom Company

An Architect's Guide to Microservices: Accelerate Microservices and API Development with Tools from Layer7

Layer7 provides the proven platform for a scalable, secure microservices solution for the enterprise.

Challenge

In the application (app) economy, many enterprises must rebuild applications that need to quickly adapt to changing needs. The traditional way of rolling out and supporting large applications is just not sufficient. Today, enterprise architects and VPs of applications are wondering:

- How can I deploy and release modern applications in days or weeks, not months or years and minimize downtime on app updates?
- How can I leverage multiple development teams on different language platforms to build those modern applications?
- How can I scale applications as needs change, while minimizing infrastructure costs to accommodate that scaling?

Opportunity

These challenges stem from an increased focus on agility and scale for building modern applications. Traditional application development methodology cannot support this environment.

Layer7 has expanded full lifecycle API management to include microservices, an integration that enables best-of-breed services to work together. Layer7 enables enterprises to use best practices and industry leading technology. Layer7 also provides a platform for modern architectures, and secures an environment for agility and scale. This combination enables you to accelerate the process of architecture modernization and to make it more practical.

Benefits

Layer7 provides a design, development, deployment, monitoring and management platform for your microservices architecture.

- **Live API Creator** delivers an easy–to–use solution for designing and deploying microservices, reducing your time to market for your modern applications.
- **Microgateway and API Gateway** provide the layered security and management infrastructure that a microservices architecture demands, ensuring that your enterprise IP remains secured.
- **OAuth Toolkit**, a component of API Gateway, enables authorization as well as integration with identity and access management solutions. OAuthToolkit improves CX for app users, while extending enterprise security from end to end.
- **Application Performance Management™** (APM) monitors your containerized systems, proactively ensuring application availability and a good CX.

Solution

Microservices and Digital Transformation

This solution brief describes how microservices can provide the agility that enterprises need to meet the demands of the app economy. Digital transformation is not a walk in the park. Indeed, building modern application architectures can be daunting to set up and implement. To help you face those challenges, we will answer the following questions:

- What are microservices?
- Why are they so important?
- Do I really need microservices?
- How do I enable microservices?
- What is the connection between microservices and APIs?
- How do I secure microservices?

What are Microservices?

Monolithic applications are no longer efficient in the digital economy. Development teams across enterprises are struggling to develop, deploy and scale large applications. Microservices came about due to the widespread adoption of the DevOps frameworks and agile methodologies. Development teams felt the pinch to break down complex application silos into simpler code blocks, which gave birth to microservices.

Architects devised design patterns to turn complex applications into simple and fine-grained yet reusable and interoperable processes that can be modified and deployed independently of each other. These fine-grained processes are called microservices.

Microservices Components

The API Academy has defined some common characteristics of a microservices architecture. The industry has adopted the following standards to characterize a microservice.

- Small size
- Messaging-enabled
- Bounded by contexts
- Autonomously developed
- Independently deployable
- Decentralized
- Language-agnostic
- Built and released with automated processes

While you may see the terms *microservice* and *microservice architecture* used interchangeably, they are not quite the same. A microservice architecture is made by engineering highly-automated, evolvable software systems that contain capability-aligned microservices.

As a rule, these components are deployed in a Docker container. The Docker container provides a centralized runtime environment, although you can have many Docker containers. Another benefit of using Docker containers is that it somewhat eases the management burden.

Solution (cont.)

Why are Microservices So Important?

Every digital enterprise is trying to thrive in the digital economy. And all of these enterprised aspire to two things: speed and scale. A company's need to get to market faster is critical; and it is equally important to be able to scale up appropriately to support the increasing customer demand. The key mantra here is: speed and safety at scale. You can only succeed when you attain speed and scale without losing safety.

Agile and DevOps models support decentralized and distributed ownership of software assets and promote faster turnaround of changes and quick deployment. However, to intelligently break down complex, monolithic applications into autonomous units, you need a design strategy, namely, microservices. By breaking your huge application into microservices, you enable your development team to be more nimble with updates and autonomous deployments. This removes dependencies to create large and complex builds. It also eliminates the need for over-sophisticated architectures to step up the scale to meet volume demands.

Can My Enterprise Benefit from Microservices?

Every enterprise has different needs at different times. There are definitely times when microservices can provide a solution, and times when a monolithic application might be more appropriate.

First, we will take a look at the benefits and disadvantages of monolithic applications. Monolithic applications generally share some positive characteristics:

- They are easier to build, initially.
- They are easier to test against, initially.

However, monolithic applications also share some drawbacks:

- They are more difficult to manage, compared to microservices, in an integrated development environment (IDE).
- They are difficult to maintain as your codebase grows.
- They are slower to iterate as time goes on, due to their growing codebase.
- They are difficult to scale. Infrastructure generally has to scale for the entire application, even if only one component of the application is having scalability issues.
- They are more difficult to innovate with, because of that codebase at the root.
- They are difficult to train new programmers to work with due to a steep learning curve.

Monolithic apps can make it very difficult for an organization to be agile or to scale appropriately. But, if you have an existing monolithic environment and no need for agility, then a microservices-based solution might not be a priority for you.

Solution (cont.)

If I Want to Implement Microservices, What do I Gain, and at What Cost?

Gains
Implementing a microservices architecture has several advantages in the app economy. Microservices-based applications are beneficial for the following reasons:

- They provide a better architecture for larger applications. Components can be built or swapped at will, without impacting the entire application. If something goes wrong with a component, only that component is affected.
- They are a more agile solution. It is easier to pivot segments of an application as needed.
- They are easier to learn. Each component is small and isolated, so it is a simpler process to determine what it does, how it does it, and how it interacts with the application.
- They are easier scale. Only the components that need greater scalability are affected, rather than the entire app. This ability to scale only where necessary also provides a substantial cost savings to the enterprise.

Costs
However, microservices-based applications also have their fair share of costs, both financial and otherwise.

- They contain more moving parts than a monolithic app; so it is even more important to monitor them.
- They require more modern back-end infrastructure because of their growing codebase.
- They might make it more difficult to test the complete app. But it is far easier to test individual components that make up the app.
- They require more security because of their multiple endpoints.

As a rule, for agile enterprises on a digital transformation journey, the advantages of microservices far outweigh the disadvantages.

How Do I Enable Microservices within My Enterprise?

The steps below describe one possible path that you might follow to enable microservices for an enterprise.

- Assess the maturity of the agile enterprise. If your organization is agile and you are thinking of or have adopted DevOps, your enterprise is quite ready for microservices.
- Create smaller groups of developers. Empower smaller teams of developers to own and work effectively on a smaller set of services and APIs. This inherently encourages loose coupling and autonomous deployments.
- Adopt a domain-driven design. Break down large applications into simpler services based on business capabilities or functions. The more fine-grained the services are, the better they work for this design.

Solution (cont.)

What is the Connection Between APIs and Microservices?
Microservice components only become valuable when they can communicate with other components in the system; they each have an interface or API. Just as we need to achieve a high level of separation, independence, and modularity for our code, we need to make sure that our APIs, the component interfaces, are also loosely coupled. Otherwise, you would not be able to deploy two microservices independently, which should be one of your primary goals to balance speed and safety.

An API layer in front of microservices can facilitate the support for client-side applications, such as mobile apps, because it isolates the fine-grained microservice from an app. This API layer is ideal for doing microservice orchestration and applying security.

How Do I Secure Microservices?
In virtually all microservice implementations, teams secure API endpoints, provided by microservices, with an API gateway. Modern API gateways provide additional, critical features required by microservices: transformation and orchestration. Finally, in most mature implementations, API gateways cooperate with service discovery tools to route requests from microservices clients. A microservice architecture is one with a significantly high degree of freedom.

In mature microservices organizations where the architecture is implemented for complex enterprise applications, it is common to deploy hundreds of microservices. In these cases, security is a very critical factor to consider. APIs provided by microservices might call each other, might be called by front-end, public-facing APIs, or might be directly called by API clients, such as mobile applications, web applications, and partner systems. The widely recommended approach is to secure invocation of public-facing API endpoints of the microservices-enabled system using a capable API gateway coupled with an OAuth provider. An API gateway is a key component of any microservices architecture and acts as a common bridge between the service implementation and any consuming clients. API gateways provide the following benefits:

- Centralized security enforcement for authentication, authorization and threat protection.
- Routing and mediation to protected resources across various protocols.
- Service level management for enforcing business-level rate limits and quotas.
- Service orchestration for reducing service invocations.
- Service interfaces for exposing application-specific interfaces from monolithic back ends.

Tools

Tools from Layer7 for Microservices

Microservices architecture has many moving parts and all of those parts must work together seamlessly. Layer7 views this in a holistic fashion, and offers full lifecycle API management as the platform that provides a complete solution.

Live API Creator is an automated, low-code alternative development solution from CA. Live API Creator creates and exposes domain-driven microservices and the REST or JSON APIs of apps. These microservices and APIs provide access to existing data and functionality from both legacy and modern data sources and apps. The solution enables developers, using a point-and-click approach, to create new REST endpoints that join data across diverse data sources. API owners can extend the API with declarative business rules, JavaScript event processing,role-based security, and interactive testing.

Live API Creator also enables companies that have embraced API management to expand the scope of their API lifecycle. You can move beyond management and enforcement in existing gateway and portal offerings toward the creation of APIs closer to the data layer. With Live API Creator, you can rapidly create application back-ends for internal applications, mobile development projects, data-as-a-service exposure, Internet of Things (IoT) enablement, and partner integration.

From a microservices perspective, the solution addresses some key objectives:

- **Modularity**. Use Live API Creator to decompose large applications into self-contained units called resources. Resources deliver everything necessary for app delivery, data integration, business logic and a robust API interaction layer. Resources are message-based, RESTful APIs that are independent of the underlying schema.
- **Speed of delivery**. Resource definition is point and click, integrating multiple objects from multiple databases.
- **Zero deployment**. The solution eliminates the delays associated with deployment. Defined resources are immediately executable as soon as you click save; there is no need to compile or deploy.
- **Automated deployment**. Alternatively, you can export a microservice and employ scripts to import it to production.
- **Cohesion**. Dependencies are automated, so deploying one microservice does not affect others.
- **Separation of concerns**. Live API Creator separates microservice creation from business logic, which is defined on underlying domain.

Microgateway is a lightweight, containerized gateway, that is designed to scale within highly decentralized environments. Microgateway supports common microservices patterns by providing service discovery, routing, rate limiting, last-mile security, and local aggregation and orchestration. Microgateway is also easily deployable and configurable by developers at design time using provided policy templates.

Tools (cont.)

Microgateway integrates with industry-standard DevOps tools for scripted production deployments and can be extended to support custom or new use cases by creating new templates and baking them into new pre-configured containers.

Microgateway enables developers to embrace new patterns as they emerge within microservices environments and provides the traffic management and mediation necessary for microservice architectures, large and small. And lastly, it provides the security and fault tolerance necessary for regulated industries.

API Gateway, including Essentials, Enterprise, and Layer7 Mobile API Gateway, delivers industry–leading gateway functionality for enterprise-class microservices. API Gateway does this by combining policy management with runtime policy enforcement and by delivering a central policy enforcement point between the business and the end user, no matter where they are located.

With API Gateway, enterprises can selectively open their data and applications to both internal and third–party developers, integrating with existing identity and access management (IAM) solutions for a plug–and–play solution. API Gateway deploys in a variety of form factors including Docker, which is ideal for microservices because it easily scales and can be deployed in a failover environment for high availability. The solution also includes protocol-bridging, providing full translation between a variety of protocols, from legacy to REST and JSON, and from legacy to mobile, cloud and social.

For a microservices architecture deployment perspective, Microgateway and API Gateway address some key objectives:

- **Security**. The solution can act as the central policy enforcement point. It is better to always secure any API or microservice access with an API gateway, and in most cases, the negligible overhead of introducing an API gateway in between service calls is well worth the benefits.
- **Transformation and orchestration**. API Gateway allows you to declaratively, through configuration, create API interfaces. These API interfaces can orchestrate back–end microservices and hide their granularity behind a much more developer friendly interface to eliminate chattiness.
- **Routing**. API Gateway hides the complexities of routing to a microservice from client apps. The solution can interface with either HTTP or DNS interfaces of a service discovery system. The solution routes an API client to the correct service when an external uniform resource identifier (URI) associated with a microservice is requested.

Microgateway is ideally suited for deployment. The solution scales along with the microservices it manages, within the same PaaS and container management environments. Microgateway is often coupled with API Gateway, deployed at the edge of the application tier or network, and integrated with existing infrastructure such as OAuth servers and centralized logging or auditing systems. API Gateway is ideally coupled with Microgateway in microservice environments, but also applies to traditional monolithic architectures.

Tools (cont.)

OAuth Toolkit runs on top of industry best–of–breed API Gateway. OAuth Toolkit provides an OAuth provider and token management system to control access to microservices from web, mobile and other applications. OAuth Toolkit allows you to deliver these OAuth provider functions by extending your existing identity infrastructure and is highly scalable. The solution includes the following features:

- An OAuth authorization server for issuing access tokens in both two– and three–legged OAuth flows.
- An OAuth resource server for API access control and policy enforcement.
- Customizable templates for OAuth client and user implementations.
- Integration with all popular identity and access management (IAM) and single sign–on (SSO) solutions.
- The ability to bridge between OAuth and other access control standards.
- The ability to choose between token types such as a JSON Web token (JWT).
- The ability to implement custom handshakes for tailored user experiences.

Using the OAuth Toolkit, you can create a distributed authentication mechanism for microservices, ensuring a secure solution.

Traditional static topology mapping and instrumentation is best suited for monolithic systems and has less relevance for microservices. Application Performance Management (APM) provides a unique architecture to manage dynamic microservices and the ephemeral nature of containers. APM employs a radical, future–proofing approach to managing containerized systems. APM includes simplified configuration and visibility into modern system complexity, especially microservice interdependencies and communication flows.

APM for microservice architectures is a multifaceted monitoring solution. As a foundational service, agentless monitoring automates the discovery of containers and dependencies. Agentless monitoring immediately surfaces key health indicators, such as CPU saturation, error rates, and latency. A powerful service in itself, this solution is further enhanced by the automated capture of container attributes and a data model that enables microservice performance to be viewed from multiple perspectives. This approach is well matched to microservices architectures. Using APM, engineers can quickly and easily distil complex topologies into service views where performance is automatically aggregated.

Next Steps

In many cases, container monitoring will need to be enriched with application-centric performance indicators. APM supports this by enabling application instrumentation within containers. Using this instrumentation, you can access advanced application performance services in the context of supporting microservice architectures. For example, you might use statistical techniques to manage performance baselining and reduce alert noise. Or, you might you use transaction tracing and assisted triage to gather detailed evidence and build remediation workflows.

Agentless, container-centric monitoring and deeper application instrumentation are valuable services in themselves. However, APM goes further, delivering higher-level insights by combining the information that these services expose. By automatically correlating application performance to container health, APM not only provides DevOps teams with exact root cause indicators for problems but also details which container-application configurations deliver the best possible performance.

Today’s DevOps and agile-loving enterprises are striving for fast changes and quick deployments. To these companies, the microservices architecture is a boon, but not a silver bullet. Organizations can enable smaller development teams with more autonomy and agility, and as a result, the business will notice IT being more in tune with their changing demands.

IT will need to align its API strategy with the microservices that developers produce. Securing those microservices should be of the utmost importance; leveraging API Gateways in this context will benefit IT. And always remember, that if you are looking for speed and scale, safety and a strong management component is equally important.

To learn more about microservices, and Layer7 API management, please download our e-book: Microservice Architecture: Aligning Principles, Practices, and Culture.

For more product information, please visit ca.com.

ca technologies A Broadcom Company

Broadcom Inc. (NASDAQ: AVGO) is a global technology leader that designs, develops, and supplies a broad range of semiconductor and infrastructure software solutions. Broadcom’s category-leading product portfolio serves critical markets including data center, networking, software, broadband, wireless, storage and industrial. For more information, go to www.broadcom.com. As of November 2018, CA Technologies was acquired by Broadcom Inc.