An Architect's Guide to Microservices

Accelerate microservices and API development with tools from CA Technologies.

ca technologies

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

Executive Summary

Challenge

Faced with the app economy, many enterprises must rebuild applications that need to quickly adapt to changing needs; and the traditional way of rolling out (and supporting) large applications just isn't sufficient. Today's 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—and traditional application development methodology cannot support this environment.

CA Technologies has expanded full lifecycle API management to include microservices—an integration enabling the best of breeds to work together to provide the platform for modern architectures and a secure environment for agility and scale. CA enables enterprises to use best practices and industry-leading technology to accelerate and make the process of architecture modernization more practical.

Benefits

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

- **CA Live API Creator** delivers an easy-to-use solution for designing and deploying microservices, reducing your time to market for your modern applications
- **CA Microgateway and CA 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 CA API Gateway, enables authorization as well as integration with identity and access management solutions, improving CX for app users, while extending enterprise security end to end.
- **CA Application Performance Management** monitors your containerized systems, proactively ensuring application availability and a good CX.

Section 1:

Microservices and Digital Transformation

This paper describes how microservices can provide the agility enterprises need to meet the needs of the application economy. Digital transformation is not a walk in the park; indeed, building modern application architectures can be daunting to set up and implement. We'll answer questions including:

- What are microservices?
- Why are they so important?
- Do I really need microservices?
- How do I enable microservices?
- What's 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 deal with large applications with respect to development time, deployment and scalability. With the widespread adoption of 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 have 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 sees certain common characteristics of a microservices architecture, and the industry has adopted those characteristics as the standard. With that in mind, a microservice has these important characteristics:

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

While you may see microservice and microservice architecture used interchangeably, they're not quite the same. A microservice architecture is a style of engineering highly-automated, evolvable software systems that are made up of capability-aligned microservices.

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

Why are microservices so important?

Every digital enterprise trying to thrive in the digital economy is aspiring for two things: speed and scale. If a company's need to get to market faster is critical, it's equally important to be able to scale up appropriately to support increasing customer demand. But 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're enabling your development team to be more nimble with updates and autonomous deployments. This removes dependencies to create large and complex builds, and it eliminates the need for over-sophisticated architectures to step up scalability to meet volume demands.

Can my enterprise benefit from microservices?

Every enterprise has different needs—there are definitely times when microservices are a good pattern to provide a solution, and times when a monolithic application may be more appropriate.

Monolithic applications are:

- Generally easier to build, initially
- They (again, generally) are easier to test against, initially
- If you're using an integrated development environment (IDE), there may be a better fit

However, monolithic applications also are:

- Difficult to maintain as your codebase grows
- Slower to iterate as time goes on, due to the growing codebase
- Difficult to scale; infrastructure generally has to scale for the entire application, even if it's only one component of the application that's having scalability issues
- More difficult to innovate with, because of that codebase at the root
- Difficult to insert new programmers against due to a steep learning curve (and that codebase at the root)

This 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 may not be important.

If I want to implement microservices, what do I gain, and at what cost?

Implementing a microservices architecture has several advantages in the app economy.

Microservices–based applications are:

- A better architecture for larger applications: components can be built/swapped at will (without impacting the entire application). Also, if something goes wrong, only that component is affected.
- A more agile solution: it's easier to pivot segments of an application as needed.
- Easier to learn: each component is small and isolated, so it's a simpler process to determine what it does, how it does it and its interaction with the application.
- Easier to scale: only the components that need improved scalability are affected, rather than the entire app (also providing a substantial cost savings to the enterprise).

However, microservices–based applications also have:

- More moving parts than a monolithic app, so it's important to monitor
- More modern back–end infrastructure requirements to the growing codebase
- More difficulty testing the complete app (but it's far easier to test individual components)
- Security requirements that need to be addressed on behalf of endpoints

For enterprises on a digital transformation journey, with the need to be agile, the advantages of microservices far outweigh the disadvantages, as a rule.

How do I enable microservices within my enterprise?

Assess the maturity of the agile enterprise. If your organization is agile and you're 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/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.

What's 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 of our code, we need to make sure that our APIs, the component interfaces, are also loosely coupled. Otherwise, you won't 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) because it isolates the fine–grained microservice from the app. This layer is ideal for doing microservice orchestration and applying security.

How do I secure microservices?

A common pattern observed in virtually all microservice implementations is teams securing 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's common to have hundreds of microservices deployed. Security will be a very critical factor to consider in that case. In virtually all microservice implementations, we see API endpoints provided by various microservices that are secured using a capable API gateway. APIs provided by microservices may call each other, may be called by front-end, (public-facing) APIs, or may 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:

- 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 façades for exposing application-specific interfaces from monolithic back ends.

The gateway applies security on behalf of microservices. However, there's also typically an OAuth provider that manages security sessions and works side by side with the API gateway.

Section 2:

Tools From CA Technologies for Microservices

As noted earlier, a microservices architecture has **many** moving parts, and all must work together seamlessly. CA Technologies views this in a holistic fashion, and offers full lifecycle API management as the platform that provides a complete solution.

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

role-based security and interactive testing. CA Live API Creator also enables companies that have embraced API management to expand the scope of their API lifecycle beyond management and enforcement in existing gateway and portal offerings toward the creation of APIs closer to the data layer. With CA 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 CA Live API Creator to decompose large applications into self-contained units called resources, delivering 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 deploy:** The solution eliminates the delays associated with deployment. Defined resources are immediately executable as soon as you click save—no compile, no deploy.
- **Automated deploy:** Alternatively, you can export a microservice (e.g., from development) and employ scripts to import it (e.g., to production).
- **Cohesion:** Dependencies are automated, so deploying one microservice does not affect others.
- **Separation of concerns:** CA Live API Creator separates microservice creation from business logic, which is defined on underlying domain.

CA Microgateway is a lightweight, containerized gateway, designed to scale within highly decentralized environments. It supports common microservices patterns by providing service discovery, routing, rate limiting, last mile security, and local aggregation and orchestration, and is easily deployable and configurable by developers at design time using provided policy templates. It integrates with industry-standard DevOps tools for scripted production deployments and can be extended to support custom/new use cases by creating new templates and baking them into new pre-configured containers.

CA 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.

CA API Gateway (including Essentials, Enterprise, and CA Mobile API Gateway) delivers industry-leading gateway functionality for enterprise-class microservices by combining policy management with runtime policy enforcement and delivering a central policy enforcement point between the business and the end user—no matter where they're located. With CA 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. CA 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 the bridge from legacy to mobile, cloud and social.

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

- **Security:** As mentioned above, the solution can act as the central policy enforcement point. It's better to always secure any API/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:** CA API Gateway allows you to declaratively, through configuration, create API interfaces that can orchestrate back-end microservices and hide their granularity behind a much more developer friendly interface to eliminate chattiness.
- **Routing:** CA API Gateway hides the complexities of routing to a microservice from the client apps. The solution can interface with either HTTP or DNS interfaces of a service discovery system and route an API client to the correct service when an external URI associated with the microservice is requested.

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

OAuth Toolkit from CA runs on top of industry best-of-breed CA 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:

- 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 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 JWT
- The ability to implement custom handshakes for tailored user experiences

By utilizing OAuth Toolkit from CA, you can create a distributed authentication mechanism for microservices, ensuring a secure solution.

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

CA APM for microservice architectures is a multifaceted monitoring solution. As a foundational service, agentless monitoring automates the discovery of containers and dependencies, immediately surfacing 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, since engineers can quickly and easily distil complex topologies into service views where performance is automatically aggregated.

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

Agentless container-centric monitoring and deeper application instrumentation are valuable services in themselves, but CA APM combines the information they expose to deliver higher-level insights. By automatically correlating application performance to container health, CA APM not only provides DevOps teams with exact problem root cause indicators but details which container-application configurations deliver the best possible performance.

Section 3:

Next Steps

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're looking for speed and scale, safety is equally important—and a strong management component is a must.

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

To learn more about microservices and CA API Management visit ca.com/microservices

Connect with CA Technologies at ca.com

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate – across mobile, private and public cloud, distributed and mainframe environments. Learn more at ca.com.

CS200-304503_0917